

UNIVERSITY OF PATRAS  
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT  
ELECTRONICS AND COMPUTERS SECTOR



# Design and development of a mobile editor for location-based games

---

TaggingCreaditor Application

**Rodrigo López-Romero Guijarro**

STUDENT OF THE HIGHER TECHNICAL SCHOOL OF TELECOMMUNICATIONS ENGINEERING  
(UNIVERSITY OF VALLADOLID)

SUPERVISORS: PROF. N.AVOURIS

**DIPLOMA THESIS NUMBER:**

**JULY 2014**

To my parents, sisters and family for their unconditional support and education received along all these years, because I have managed to fulfil a dream; to Andrea, for her love and all these amazing years together; to Alejandro, for these four years together working hard; to Christos, Menita and Prof. Avouris, for all the help I have received during the course of the project; to all the people who helped during the evaluation of the application; and last but not least, to all the wonderful people I have met in Greece.

Honestly, thank you so much!



## CERTIFICATION

It is certified that diploma thesis with the title:

### DESIGN AND DEVELOPMENT OF A MOBILE EDITOR FOR LOCATION-BASED GAMES

by the student of the University of Valladolid

---

López-Romero Guijarro,  
(Surnames)

---

Rodrigo  
(Name)

Was presented in public at the department of Electrical and Computer  
Engineering of the University of Patras, Greece on July 10th, 2014.

The supervisor

The Head of the Electronics and Computers Division

N. Avouris

E. Housos





## ABSTRACT

In the present project I propose to describe the design, development and implementation of mobile application TaggingCreaditor. TaggingCreaditor, as its name suggests, is a tool that allows the creation and edition of content for collaborative location-based mobile games, i.e., for the called 'Tagging games', whose physical elements are identified with tags such as a QR code or RFID code tags. The application has been developed for both desktop computers and mobile devices. In this paper the editor for the mobile devices will be presented.

Mobile technologies play an increasingly important role in our lives, so that it is reasonable that new ideas come up to take advantage of this technology. An interesting example is the collaborative location-based mobile games that have the great advantage that they can be used for both entertainment and learning. Examples of collaborative location-based games are MuseumScrabble or CityScrabble, of which a study will be carried out prior to the design of the content editor.

The problem is that there are plenty of locations where you can play many of the existing location-based games, but we do not have a generic tool that facilitates the creation of content. This is what comes to solving TaggingCreaditor: creating content for these games so that any developer can operate his game at any location in the world. Even, a school teacher could use this tool to create content for their students in a museum to which they will visit. Therefore, this application could be considered as a useful supporting utility for collaborative location-based mobile games such as MuseumScrabble or CityScrabble mentioned before.

Among the highlights and strengths of the application, we find that it is a generic and collaboration tool. First, the application is designed with the idea that it can be used by a large number of mobile location-based games, so it is a generic tool. And, secondly, the content editor has the advantage of being collaborative, allowing multiple users to create content for their own games in parallel in the same database.

I would like to finish remarking the utility that a generic editor may have for the location-based games, allowing to create content of any subject for whatever the game is and in any place of the world. I also would like to highlight that the editor can be improved in a lot of different ways, for instance, an interesting improvement could be the multi-user support so that each user can create their own games without using the same database that other users.

## Περίληψη

Η παρούσα εργασία περιγράφει το σχεδιασμό, την ανάπτυξη και την υλοποίηση της εφαρμογής TaggingCreaditor για κινητές συσκευές. Το TaggingCreaditor όπως αποκαλύπτει και το όνομα του είναι ένα εργαλείο για την παραγωγή και επεξεργασία περιεχομένου για συνεργατικά χωρο-ευαίσθητα παιχνίδια με κινητές συσκευές, όπως για παράδειγμα τα “Tagging games” όπου τα φυσικά αντικείμενα αναγνωρίζονται μέσω της σάρωσης QR ή RFID ετικετών. Η εφαρμογή αναπτύχθηκε για χρήση σε σταθερό υπολογιστή αλλά και σε κινητές συσκευές. Σε αυτή τη διπλωματική εργασία θα παρουσιαστεί η εφαρμογή που αφορά στην παραγωγή και επεξεργασία περιεχομένου για κινητές συσκευές.

Οι κινητές τεχνολογίες διαδραματίζουν όλο ένα και σημαντικότερο ρόλο στη ζωή μας, με αποτέλεσμα πολλές νέες ιδέες να προκύπτουν λαμβάνοντας υπόψη τα πλεονεκτήματα αυτών των τεχνολογιών. Ενδιαφέρον είναι το παράδειγμα των συνεργατικών χωρο-ευαίσθητων παιχνιδιών τα οποία μπορούν να χρησιμοποιηθούν παράλληλα για ψυχαγωγία και μάθηση. Τέτοια παραδείγματα αποτελούν τα παιχνίδια MuseumScrabble και CityScrabble, στα οποία έχει γίνει εκτενής μελέτη πριν το σχεδιασμό του TaggingCreaditor.

Τα υπάρχοντα χωρο-ευαίσθητα παιχνίδια μπορούν να παιχτούν σε πολλούς χώρους ωστόσο δεν υπάρχει ένα εργαλείο γενικής εφαρμογής το οποίο να διευκολύνει στην δημιουργία περιεχομένου για τα παιχνίδια αυτά. Αυτό το κενό έρχεται καλύψει η παρούσα διπλωματική εργασία με το σχεδιασμό και την ανάπτυξη του εργαλείου TaggingCreaditor. Το εργαλείο αυτό αναπτύχθηκε για την παραγωγή και επεξεργασία περιεχομένου για χωρο-ευαίσθητα παιχνίδια έτσι ώστε κάθε σχεδιαστής παιχνιδιού να μπορεί να σχεδιάσει παιχνίδια για να παιχτούν σε οποιαδήποτε περιοχή του κόσμου. Για παράδειγμα, ένας δάσκαλος θα μπορούσε να σχεδιάσει ένα παιχνίδι και να παράξει περιεχόμενο για αυτό ώστε να παιχτεί από τους μαθητές του κατά την διάρκεια μιας επικείμενης επίσκεψης σε ένα μουσείο. Επιπρόσθετα, αυτό το εργαλείο θα μπορούσε να αποτελεί υποστηρικτική λειτουργία για τα υπάρχοντα συνεργατικά χωρο-ευαίσθητα παιχνίδια όπως τα MuseumScrabble και CityScrabble.

Ανάμεσα στα κυριότερα σημεία και δυνατότητες αυτής της εφαρμογής, είναι το γεγονός ότι το TaggingCreaditor είναι ένα συνεργατικό και γενικής εφαρμογής εργαλείο. Κατά πρώτον, ο σχεδιασμός έγινε έτσι ώστε πολλοί χρήστες να μπορούν συνεργατικά και παράλληλα να παράγουν και να επεξεργάζονται το περιεχόμενο των παιχνιδιών τους, χρησιμοποιώντας μια κοινή βάση δεδομένων. Κατά δεύτερον, η εφαρμογή αυτή σχεδιάστηκε με το σκεπτικό πως θα μπορεί να χρησιμοποιηθεί για την παραγωγή και επεξεργασία του περιεχομένου ενός μεγάλου αριθμού χωρο-



ευαίσθητων παιχνιδιών, κάτι που κάνει το συγκεκριμένο εργαλείο να είναι εργαλείο γενικής εφαρμογής.

Θα ήθελα να κλείσω τονίζοντας την χρησιμότητα που μπορεί να έχει ένα τέτοιο εργαλείο γενικής εφαρμογής για τα χωρο-ευαίσθητα παιχνίδια, επιτρέποντας την δημιουργία και επεξεργασία περιεχομένου γύρω από οποιαδήποτε θεματική ενότητα, για οποιοδήποτε παιχνίδι και σε οποιαδήποτε περιοχή του κόσμου αυτό παίζεται. Θα ήθελα επίσης να τονίσω πως το συγκεκριμένο εργαλείο μπορεί να βελτιωθεί με πολλούς διαφορετικούς τρόπους όπως για παράδειγμα να μπορεί να υποστηρίζει πολλούς χρήστες οι οποίοι όμως δεν θα χρησιμοποιούν κοινή βάση δεδομένων για την παραγωγή και επεξεργασία του περιεχομένου των παιχνιδιών τους.

## RESEÑA

En el presente proyecto propongo describir el diseño, desarrollo e implementación de la aplicación móvil TaggingCreaditor. TaggingCreaditor, como su nombre sugiere, es una herramienta que permite la creación y la edición de contenido para juegos móviles colaborativos basados en localización, es decir, para los llamados ‘Tagging games’, cuyos elementos físicos son identificados con etiquetas como códigos QR o etiquetas RFID. La aplicación ha sido desarrollada para ordenadores de sobremesa y dispositivos móviles. En este documento será presentado el editor para los dispositivos móviles.

Las tecnologías móviles juegan un rol cada vez más importante en nuestras vidas, de modo que es razonable que nuevas ideas surjan para aprovechar esta tecnología. Un ejemplo interesante son los juegos móviles colaborativos basados en localización que tienen la gran ventaja que pueden ser usados tanto para la diversión como para el aprendizaje. Ejemplos de este tipo de juegos son MuseumScrabble o CityScrabble, de los que se hará un estudio posterior junto con otros juegos antes del diseño del editor.

El problema es que hay muchas localizaciones donde se pueden jugar muchos de los juegos basados en localización existentes, pero no tenemos una herramienta genérica que facilite la creación de contenido. Esto es lo que trata de resolver TaggingCreaditor: crear contenido para estos juegos de forma que cualquier desarrollador pueda poner en funcionamiento su juego en cualquier localización del mundo. Incluso un profesor de colegio podría usar este editor para crear contenido para sus alumnos en un museo al que vayan a hacer una excursión. Por tanto, esta aplicación podría ser considerada como una utilidad de apoyo muy útil para los juegos basados en localización tales como MuseumScrabble o CityScrabble, mencionados anteriormente.

Entre los puntos más destacados de la aplicación, se puede decir que es una herramienta genérica y colaborativa. Primeramente, la aplicación está diseñada con la idea de que pueda ser usada por un amplio número de juegos basados en localización, por eso se considera una herramienta genérica. Y, en segundo lugar, el editor de contenido tiene la ventaja de ser colaborativo, permitiendo que múltiples usuarios creen contenido para sus propios juegos al mismo tiempo y en la misma base de datos.

Me gustaría terminar remarcando la utilidad que un editor genérico puede tener para los juegos basados en localización, permitiendo crear contenido de cualquier tema para cualquier juego y en cualquier lugar del mundo. También me gustaría destacar que el editor puede ser mejorado de muchas formas, por ejemplo, una interesante mejora podría ser el soporte multiusuario de modo que cada usuario pueda crear contenido para sus propios juegos sin la necesidad de usar la misma base de datos que otros usuarios.



---

**TABLE OF CONTENTS**

<b>1. INTRODUCTION .....</b>	<b>16</b>
1.1. Motivation.....	17
1.2. Objectives .....	18
1.3. Mobile location-based games.....	18
1.3.1. MuseumScrabble .....	19
1.3.2. CityScrabble .....	20
1.3.3. Benaki Museum Scrabble .....	21
1.3.4. Taggling.....	22
1.4. Thesis organization .....	23
<b>2. TECHNOLOGY .....</b>	<b>24</b>
2.1. Programming languages .....	25
2.1.1. Php.....	26
2.1.2. Html .....	27
2.1.3. JavaScript.....	28
2.1.4. Css.....	29
2.1.5. Sql .....	30
2.1.6. Ajax .....	30
2.2. ZURB Foundation .....	31
2.2.1. A little backstory of Foundation .....	31
2.2.2. Features .....	32
2.3. MySQL Workbench .....	33
2.3.1. A little backstory of MySQL .....	33
2.3.2. Features .....	33
2.4. Apache web server .....	35
2.5. PhpMyAdmin .....	36
2.6. Conclusion about the tools.....	37
<b>3. DESIGN.....</b>	<b>38</b>
3.1. Introduction .....	39
3.2. Requirements.....	39
3.2.1. Client-Server Architecture .....	39
3.2.2. Web based application .....	40

3.2.3.	Centralized database .....	40
3.2.4.	Connection to maps.....	40
3.2.5.	Desktop & mobile application .....	41
3.2.6.	Different terminology support .....	41
3.3.	Comparison of location-based games .....	41
3.4.	Abstractions: the first approach .....	43
3.4.1.	Real world abstractions .....	44
3.4.2.	Virtual world abstractions .....	45
3.4.3.	Relationship between the two worlds .....	46
3.4.4.	Summary.....	46
3.5.	Design of the entities hierarchy: final design .....	47
3.5.1.	Game entity .....	49
3.5.2.	Real world entities.....	49
3.5.3.	Virtual world entities .....	50
3.6.	User interface design .....	51
<b>4.</b>	<b>APPLICATION - DEVELOPMENT AND IMPLEMENTATION .....</b>	<b>56</b>
4.1.	General aspects.....	57
4.1.1.	Phases and organization of the project.....	57
4.1.2.	Differentiation between mobile and desktop.....	59
4.2.	Database design.....	60
4.2.1.	Game attributes.....	63
4.2.2.	Attributes of the real world entities.....	66
4.2.3.	Attributes of the virtual world entities.....	68
4.2.4.	Relationship Element-Key.....	70
4.3.	User interface implementation .....	71
4.3.1.	Main screens.....	71
4.3.2.	Navigation through the main screens .....	76
4.4.	Usage scenarios .....	77
4.4.1.	Mobile example .....	77
<b>5.</b>	<b>EVALUATION AND RESULTS .....</b>	<b>80</b>
5.1.	Introduction .....	81
5.2.	Before the evaluation .....	82

5.2.1. Documents .....	83
5.2.2. Material .....	84
5.3. Think-Aloud Protocol .....	84
5.3.1. Advantages of Think-Aloud Protocol .....	85
5.4. SUS – System Usability Scale .....	85
5.4.1. Operation.....	85
5.4.2. Scoring and interpreting SUS.....	86
5.5. Focus group.....	87
5.6. Evaluation session.....	88
5.6.1. Initial information phase .....	88
5.6.2. Description of location-based games .....	89
5.6.3. Scenario description .....	89
5.6.4. Demonstration of TaggingCreaditor use .....	89
5.6.5. Action phase .....	90
5.6.6. SUS Questionnaire .....	90
5.6.7. Focus group discussion .....	91
5.7. Results and conclusions .....	91
5.7.1. Results during the evaluation.....	91
5.7.2. Analysis of SUS questionnaires.....	92
5.7.3. Analysis of the focus group .....	95
<b>6. CONCLUSIONS .....</b>	<b>97</b>
<b>7. REFERENCES .....</b>	<b>98</b>
<b>ANNEX 1.MOBILE USER MANUAL.....</b>	<b>100</b>
<b>ANNEX 2. EVALUATION SESSION DOCUMENTS.....</b>	<b>116</b>
1. Evaluation of <i>TaggingCreaditor</i> .....	116
2. CityScrabble - Instructions for the mobile scenario .....	122
3. Taggling - Instructions for the mobile scenario .....	126

## FIGURE INDEX

Figure 1. Two main screens in MuseumScrabble: topic selection and exhibit association .....	19
Figure 2. Main screens in the CityScrabble game. ....	20
Figure 3. Main screens in Benaki Museum Scrabble game.....	21

Figure 4. Programming languages layer structure .....	25
Figure 5. Foundation is the first open-source framework to be responsive.....	32
Figure 6. MySQL Workbench provides an easy to use visual way to design databases .....	34
Figure 7. Screenshot of the database uploaded to the server through phpMyAdmin.....	36
Figure 8. The first sketch of the design which was later modified.....	44
Figure 9. First approach of the real world abstractions.....	45
Figure 10. First approach of the virtual world abstractions.....	45
Figure 11. Design of the entities hierarchy. ....	48
Figure 12. Sketch of the design of the real objects management screen.....	52
Figure 13. Sketch of the design of the virtual objects management screen.....	53
Figure 14. Sketch of the design of the linker screen. ....	53
Figure 15. Phases in which the project has been developed. ....	57
Figure 16. JavaScript script which allows to recognize a mobile device. ....	59
Figure 17. First lines of gamelist_mobile.php showing the use of sessions. ....	59
Figure 18. Sketch of the database design created before the final design.....	60
Figure 19. Database design in MySQL Workbench 6.1 CE. ....	61
Figure 20. Game entity in the database (MySQL Workbench 6.1 CE).....	63
Figure 21. Playspace entity in the database (MySQL Workbench 6.1 CE) .....	66
Figure 22. Location entity in the database (MySQL Workbench 6.1 CE) .....	66
Figure 23. Element entity in the database (MySQL Workbench 6.1 CE).....	67
Figure 24. Theme entity in the database (MySQL Workbench 6.1 CE) .....	68
Figure 25. Topic entity in the database (MySQL Workbench 6.1 CE) .....	68
Figure 26. Key entity in the database (MySQL Workbench 6.1 CE) .....	69
Figure 27. Relationship element-key entity in the database (MySQL Workbench 6.1 CE).....	70
Figure 28. Welcome screen of TaggingCreaditor. ....	71
Figure 29. Game list screen of TaggingCreaditor (left: games list, right: game details) .....	72
Figure 30. Real world management screen (left: navigation list, right: entity details boxes) ....	73
Figure 31. Virtual world management screen (left: navigation list, right: entity details boxes) .....	74
Figure 32. Easy-way linker screen in three simple steps: choose real object, choose virtual object and create relationship with the chosen values. ....	75
Figure 33. Navigation through the main screens of TaggingCreaditor. ....	76
Figure 34. Example of real content that a teacher can create for CityScrabble. ....	78
Figure 35. Example of virtual content that a teacher can create for CityScrabble. ....	78
Figure 36. Example of relationships that a teacher can create for CityScrabble. ....	79
Figure 37. Photo taken during the assessment session. ....	81
Figure 38. System Usability Scale response format. ....	86
Figure 39. Screenshot of the video tutorial showed during the evaluation. ....	89
Figure 40. A work team taking a photo of a replica while they are creating content. ....	90
Figure 41. Focus Group discussion which took place in the end of the evaluation.....	91
Figure 42. Graph showing the average rating of the SUS questionnaire for the mobile version. .....	93
Figure 43. Chart showing a comparison between the average ratings of the mobile and desktop versions. ....	94





# 1. INTRODUCTION

---

In this first chapter I will take a brief look the main aspects of the project. First of all, I will explain the motivation which led me to carry out this project and the need of a generic editor for location-based games. In the second point, I will present the main objectives and requirements that were established at the beginning of the project. After that, I will go on describing the key aspects of the location-based games along with a short explanation of each game which was taken into account for the design, development and implementation of this project. Finally, I will talk about the chapters of this thesis including a brief description of each one.

## 1.1. Motivation

In recent years, there has been a proliferation in creating new mobile games in which the user's location is of great importance. This is because the activity of the game in most of these cases evolves according to user location and events that make the player decide. The applications included in this group of games are called "location-based".

Examples of such games are called "urban games" or "street games"; multiplayer games are performed in the streets of the cities. Another group of games that can serve as an example are the museum mobile interactive games, mobile applications that, in addition to providing information about the exhibits, support more advanced interactivity with them, applying socio-cultural theories of learning, and providing visitors with tools to organize and control the supplied information. The games in first example provided can be played in many diverse places and extend to long periods of time, whereas, in the second example, the games are designed to event based. It is worth highlighting that, while the major goal of most of these games is fun and entertainment, many times, learning plays an essential role in the design of a location-based application.

There are uncountable mobile location-based games and, of course, plenty of locations in which they can be played. All these applications need to create, in one way or another, virtual content for each of the locations in which the developers wish to play, in addition to some method to label all physical elements that take part of the content of the game (examples of these methods can be QR, RFID codes ... ).

The problem is that there is no generic tool or an easy way to create or edit content for those games, but the content has to be fixed by the corresponding game developer based on location. This makes us realize the importance of generic tools that are not connected to a specific application, but a group of applications (in this case games).

The best option to solve this problem is to create a user-friendly tool that allows us to create content and can be used for any type of location-based game and in any location. This content editor will allow developers of location-based games to launch their games in a large number of locations just using this editor.

## 1.2. Objectives

The main goal is, therefore, to design and develop a generic mobile application that works as a tool for creating and editing content for location-based mobile games.

To do this, one of the first requirements to be met by the mobile editor of content will be the support of the creation of as many location-based mobile games as possible.

To achieve this, the first step will be describing several existing mobile location-based games that will be taken into account for the design of the application so that it is as abstract as possible. Having described the operation of these games, all the features they have in common and their differences will be studied.

After this, the next step is to find all possible abstractions of games proposed in order to design, then, a hierarchy of entities that will guide us in the design of the database.

Finally, once these steps are carried out and a sketch of content editor has been designed, the programming of the user interface will be started with all the necessary functionalities.

## 1.3. Mobile location-based games<sup>1</sup>

A mobile location-based game, as established before, is a type of pervasive game in which the player's location affects to the progress in the gameplay.

In this paragraph I will describe the location-based games that will be taken into account for the design and development of the generic editor to create content. Mainly, the definition, the rules and the operation of the games will be explained.

---

<sup>1</sup> References: [1] and [2]

### 1.3.1. MuseumScrabble<sup>2</sup>

MuseumScrabble is a location-sensitive and multi-player educational mobile game aimed at children visiting historical museums. The game requires that the players should explore the museum and link abstract concepts with physical artefacts using a mobile device.

The rules of the game are as follows:

The game is played by teams who compete against each other. Each team is given a single handheld device with an RFID scanner through which they link exhibits to topics (the exhibits have attached a RFID tag and the museum is equipped with a wireless network). The players select a topic and then move around the museum halls to identify the most relevant exhibits using the hints provided. The teams that identify relevant exhibits are awarded with points.

The game ends when all available links have been established and the team that has achieved the highest score when the game finishes wins the game. So, after each team has chosen a topic to play, these teams have to achieve the purpose to scan all the exposed exhibits in the museum belonging to the topic chosen.

Nowadays, the game is deployed in the historical and cultural Museum of Solomos and Eminent Zakynthians in Zakynthos, Greece, whose paintings, personal belongings and other artefacts are mainly related to prominent people of the region, from the last three centuries. It is also deployed in the museum of the University of Patras.



Figure 1. Two main screens in MuseumScrabble: topic selection and exhibit association

<sup>2</sup> References: [3] and [5]

### 1.3.2. CityScrabble<sup>3</sup>

CityScrabble is a location-based game for open areas. It is played using Android devices with 3G/Wifi connection in places for which content has been prepared. The game can be played in any open space. It can also be played in closed spaces such as museums.

The rules of the game are as follows:

Each player (or team) tries to locate locations, called also zones, or objects in space and connect them with the keys (small sentences) which appear on the screen of the device. The keys are organized in themes or concepts and one key can be connected only by a single team in a game session. In order to do that, the players have to go near them (to trigger a GPS proximity sensor) or scan a QR tag. When a location is identified, it can be connected with any key. Points are only awarded if the connection between key and the object is meaningful.

It is a competitive multiplayer game which can be played by competing individuals or by teams of 2-3 players. It should last a little more than 30 minutes, depending on the size of the area. The winner is the team with the most points when the keys are all connected.

In the following figure, we can see the main screens that are part of the game: the two first are for the starting of the game, the third one is to choose a topic, the fourth one is the screen to connect objects with keys and, finally, the last screen is the map that helps the players to locate the objects.



Figure 2. Main screens in the CityScrabble game.

<sup>3</sup> References: [4] and [6]

### 1.3.3. Benaki Museum Scrabble<sup>4</sup>

Benaki Museum Scrabble (BMS) is a location-sensitive mobile game, built for the Benaki Museum in Athens, in which each player or team has a mobile device that can read QR tags and can follow the progress of the game.

The rules of the game are as follows:

The game aims to engage visitors (mainly student groups) in a process of discovering relationships between museum exhibits and objects that are not exposed. Therefore, in the game there are two categories of objects: physical artifacts that are part of the museum's collection and are accessible to the players by means of QR tags placed near each artifact; and, on the other hand, digital objects that can be text, images or video which are part of the museum's collection, but are not present in the museum.

The game is played in rounds between players or group of players, competing with each other so that they can scan the exposed exhibits and use them through the game. Each digital object can "connect" with some of the artifacts found in the museum's halls. Therefore, relationships are created between digital and physical objects, which have as a result potential "background" stories to emerge.

Therefore, the main aim of the players is to identify one of the physical artifacts, which can be connected to the digital object that they have been given at each round, and then to make the connection between those two and gain points. The winner is the player who, at the end of the game, has managed to gain the most points.

In the following figure, the main screens of the game are shown: the first one is for the registration of a game, the following screens appear during the gameplay. The second one shows a digital object, the third one a clue about the digital object, and the last two screens appear when scanning the exhibits.

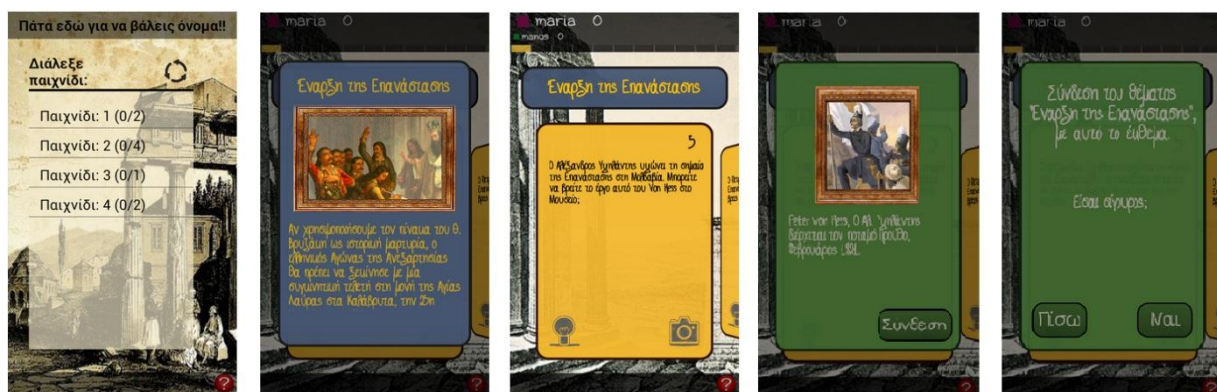


Figure 3. Main screens in Benaki Museum Scrabble game.

<sup>4</sup> References: [7]

#### 1.3.4. Taggling

Taggling, whose name comes from a mixture between tagging and tangling, is an educational location-based game which can be played in a museum, especially in contemporary art museums.

The rules of the game are as follows:

Taggling is a real-time multi-player game although it can be played by one player with a mobile device and a new approach in the eyes of contemporary art. The main purpose of the game is that the players place shuffled tags in the right order as you browse through the exhibits of a museum.

The exhibits have been described by a number of concepts, words, phrases or names that have a direct relation with the exhibit. They either characterize or describe it and so they help the player to gain some knowledge or more specific idea for the exhibit. These are the "tags" of each exhibit. Unfortunately, these tags were mingled: some jumped to other exhibits and some were hidden in the players' devices.

Next to the familiar information label of the exhibits involved in the game a small QR code which identifies it on the device of player has been added to each one. During the game, players are scanning the QR code of an exhibit to a) remove (untag) any tags they believe do not belong to the exhibit them by taking away to their device to transfer them to the right exhibit, or b) add tags from their device to the exhibit it fits best. In this way, one by one, all tags will be put back in their proper places and the game will finish. The winner is the one who has won the most points at the end of the game.

Players are awarded with points when they put the correct tags on the right exhibit (and bonus points when they complete an exhibit with all the correct tags). They lose some points when they place wrong tags.

Labels are categorized according to their importance and meaning, for example category Artist or category Technique, to indicate that the tags represent the artist or the way a work of art was made. Each category has a specific color that distinguishes it and the tags belonging to this have the same color.



---

## 1.4. Thesis organization

The rest of the report is organized as follows:

- Chapter 2 - Technology. In this chapter, it is included a description of the major aspects and features of the technology and the tools which have been used during the development of the project and the reasons I chose them.
- Chapter 3 - Design. In this third chapter I will detail all the steps that were carried out in order to design the application such as the study of the common features of the location-based games proposed.
- Chapter 4 – Application. Development and implementation. This is chapter that shows the core of the project. All the details about the development of the mobile application will be explained.
- Chapter 5 – Evaluation and results. In this chapter I will describe all the details concerning to the evaluation session of TaggingCreaditor, including the preparations, the description of the session and the obtained results.
- Chapter 6. Final conclusions.
- Chapter 7. References.
- Annexes: User guide and the evaluation session documents.



## 2. TECHNOLOGY

---

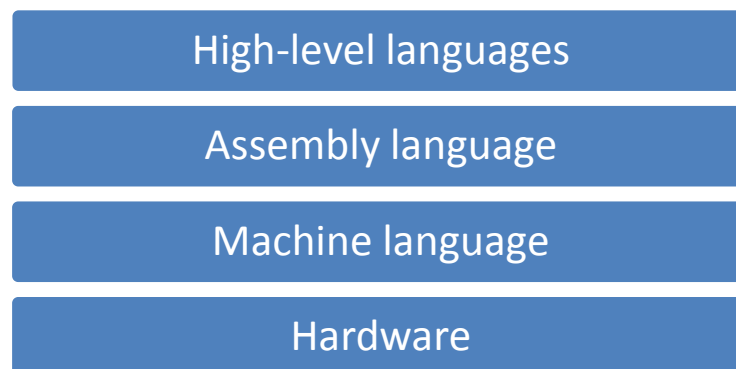
In this chapter I will describe the technology have been used to carry out this project. Firstly, the programming languages used will be explained. Afterwards, I will describe some of the frameworks I have used for the development of the project such as ZURB Foundation, MySQL Workbench or phpMyAdmin.

## 2.1. Programming languages<sup>5</sup>

It is well known by now that there is no general agreement on the meaning of the term "programming languages." Hence it is necessary to specify a general definition: A programming language is a formal language designed to express processes that can be executed by devices such as computers. They can be used to create programs that control the physical and logical behavior of a machine, to express algorithms precisely, or as a mode of human communication. It consists of a set of symbols and syntactic and semantic rules that define the structure and the meaning of its elements and expressions.

According to the classification criteria of computer programming languages, three levels may be distinguished:

- Machine languages: This language sends to the machine the fundamental operations for operation. It consists of the combination of 0's and 1's to form understandable commands for the machine hardware. Each programmable circuit, such as a microprocessor, employs its own rules and, therefore, has its own machine language.
- Assembly languages: It is derived from machine language and is made up of letters and numbers abbreviations. As a derivative of machine languages and as in the machine languages happens, each programmable circuit has its own assembly language.
- High-level languages: They are those languages that are closer to natural language than machine language. Examples of this type of languages can be Java, PHP, Python or Matlab.



**Figure 4. Programming languages layer structure**

Every processor has its own language and can execute only those programs that are written in that language. In order for the processor to understand the program written in the high-level language, some method of translation must be in place. There are two techniques to achieve this aim:

---

<sup>5</sup> References: [8] and [9].

- 1- The compiler: The compiler is a special program which creates an object program in machine code from the source program written in high-level language. In the most general sense, compilers translate from any language to any other language. The compiler treats the source program as a single unit and executes the following steps: lexical analysis (the compiler segments the source text into lexical units), syntactic analysis (to check whether syntactic rules are met), semantic analysis (to check if there are semantic errors) and code generation (to create an executable program).
- 2- The interpreter: The interpreter is a program programs whose task is adapting the instructions found in another language. Compilers and interpreters share the first three steps, but differ in the fourth one: the interpreter does not create an object program. Instead, it takes the statements (or other language elements) of the source text one after the other, interprets the statement, and executes it. We get the results immediately by having a machine code routine run.

Programming languages may rely on compilers, interpreters, or both techniques.

The question of which language is best is one that consumes a lot of time and energy among computer professionals. Every language has its strengths and weaknesses. For example, Python was designed to emphasize productivity and code readability being great for beginners because it is easy to learn, whereas PHP was designed for web development to produce dynamic web pages following a classic approach and it is more difficult to learn. It is also important to highlight that, nowadays, the majority of programmers use Integrated Development Environments (IDE) with graphical user interfaces to write programs. Such environments contain a text editor, compiler (maybe interpreter), linkage editor, loader, run time system and debugger.

Therefore and as a conclusion, the choice of a programming language depends on your knowledge of the language, the computer where the application will run on and scope of the application you are building. As a general rule, the small applications are created using a single language, and it is common to deploy large applications using multiple languages.

### 2.1.1. Php<sup>6</sup>



PHP (PHP Hypertext preprocessor) is an interpreted programming language whose commands are executed on the server and allow the creation of dynamic HTML documents. Its syntax is similar to other languages such as C, Perl, Java or JavaScript.

---

<sup>6</sup> References: [10]

PHP was designed by Rasmus Lerdorf in 1994 with the desire of creating a counter to determine the number of visits he received your online CV. However, over time it has been adopted by other developers who have transformed and become PHP in the tool is today.

PHP is an open source product, which means you can access the code, use it, modify it and distribute it for free. This is undoubtedly one of the main advantages of PHP, since the community of users developing enhancements to the system is huge.

PHP is available for many platforms including Windows, Unix or Linux and with the advantage that the applications developed in PHP can be transported from one platform to another without modifications, i.e. PHP has high portability. Although it is now a multiplatform language, it should be added that its native environment is Unix/Linux. Regarding the web server also exist different options, because it could be used for example IIS (Internet Information Server) from Microsoft or Apache, the latter with the advantage that, like PHP, is also a product of free distribution.

Among other advantages PHP also has, it allows the connection and access to multiple databases management systems like MySQL, PostgreSQL or Oracle.

Because all of the advantages I have mention in this subsection and because of the knowledge I had about PHP I chose this programming language as the main programming language to program almost all the functionalities of the application.

### 2.1.2. Html<sup>7</sup>



Along with PHP, HTML may be included as a main programming language used. It has been used, especially, embedded in PHP code and all the labels provided by HTML were used for the structure of the application.

HTML (HyperText Markup Language) is the language that all browsers use to present information on the World Wide Web (WWW) and today is one of the most important languages used to develop web pages and web applications. It is a simple language which is interpreted in the client side and which is based on the use of tags, consisting of an ASCII text enclosed within a pair of angle brackets (<>). The text included inside the brackets gives us an explanation of the usefulness of the label.

Labels may include a number of attributes, the majority of them optional, allowing us to define different possibilities and characteristics of it. These attributes will be defined by its name and the value they take.

Among other things, the management of these tags will allow us to define the logical structure of the HTML document; apply different text styles (bold, italic ...);

---

<sup>7</sup> References: [11]

inclusion of hyperlinks, which allow us to access other documents related to the current one; including images and multimedia files (graphics, video, audio); and many other features offered by HTML.

Among some of its advantages I can highlight that you don't need a particular tool to develop a program, but a text editor, such as Notepad++, to write and edit the HTML code, and a web browser which is responsible for interpreting and displaying the HTML code in your application. Despite this, it is also important to know the existence of HTML editors that can make us our work faster and can be of great help.

### 2.1.3. JavaScript<sup>8</sup>



JavaScript is one of the most used languages today, thanks to this, we have access to many features that are offered on websites we often visit and make our navigation experience much nicer, simpler and more entertaining.

JavaScript is a scripting language. A script done in JavaScript include statements or propositions that are commands given to language interpreter and can be inserted into an HTML document giving the website a degree of interactivity, because it can provide, for example, a mechanism for the detection and treatment of events as a click with the pointing device - usually a mouse.

Like HTML, JavaScript is an interpreted language by the client and, therefore, there is no restriction on the server. Each event to be processed does not require communication with the server allowing, for example, filter the information collected through a form before sending it to a database for storage.

JavaScript does not depend on other boundary conditions such as the Operating System, as already mentioned, is a language that interprets the client, so that the scripts run on all Operating System that has a ready client to do it.

JavaScript code blocks can be inserted into HTML documents in several ways. One is by using the HTML tag `<script>`, another is by using event handlers and the third by using the JavaScript URL.

Among many other things, JavaScript provides libraries created to meet very specific objectives that will help us to provide our projects with very interesting features. The amount of JavaScript libraries available are almost endless, some more complex and others more simple, but definitely, all of them are great to help when developing.

JavaScript has been used in some parts of the code in order to program some functionalities such as alert messages. It has not been used a lot because, instead of using it, I have preferred to use the library JQuery, explained in the next section.

---

<sup>8</sup> References: [12] and [13]

jQuery



jQuery is a fast, small, and feature-rich JavaScript library. It was released on January 14, 2006 and is the most used JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. There are many other JavaScript libraries, but jQuery has become the most popular due to its ease of use and great power.

jQuery is a free open source software. It has a dual-licensed under the MIT License and the GNU General Public License v2, allowing their use in free software projects and jQuery, like other libraries, offers a number of features based on JavaScript that would otherwise require a lot more code, i.e., with the functions of this library are achieved great results in less time and space. For this last reason jQuery has been used to implement a lot of functionalities in the editor application.

#### 2.1.4. Css



CSS (Cascading Style Sheets) is a style sheet language created to control the appearance or presentation of electronic documents defined with HTML and XHTML. CSS is the best way to separate content and presentation and it is essential to create complex web pages or web applications.

When creating a web application, it first uses, for instance, the HTML language for marking content, i.e., to describe the function of each element within the page: paragraph, headline, highlighted text, table, list items and so on. Once created the content, the CSS language is used to define the appearance of each element: color, size and font of text, horizontal and vertical separation between elements, position of each element on the page and so on.

Separating the definition of the content and the definition of their appearance has many advantages, as it forces us to create well-defined and complete meaning HTML documents. It also improves the accessibility of the document, reduces the complexity of its maintaining and allows us to display the same document in countless different devices. In addition to this, our web applications become more flexible, since we can change whenever we want the style of our application with only modifying our style sheets (without modifying the content); and finally, with CSS, it is important to remark the unification of the screens design of the application, since it is much easier to maintain the same appearance in all the screens.

Therefore, CSS was used during the course of the project in order to program the appearance and presentation of the application.

### 2.1.5. Sql<sup>9</sup>



SQL (Structured Query Language) is a programming language designed to store, manipulate and retrieve data stored in relational databases. SQL first appeared in 1974, when Codd (IBM) developed the first prototype of a relational database.

There is a SQL standard, but the SQL that implement major RDBMS (Relational DataBase Management System) are not exactly equal. This is due to two reasons:

- 1 - The SQL standard is quite complex, and it is impractical to implement it completely.
- 2 - Each database provider wants to differentiate their own product from the rest.

Query languages have two main parts:

- DML (Data Manipulation Language) used to get, delete, update, add ... data in the database.
- DDL (Data Definition Language) used for the creation/destruction of tables ... in the database.

This programming language has been used in order to deal with the database: to add content, to delete content, update content and so on. I chose this programming language to deal with the database because is with the one I feel more comfortable to work with.



### 2.1.6. Ajax<sup>10</sup>

AJAX (Asynchronous JavaScript and XML) is not a new programming language, but a new way of using existing tools. It was popularized by Google in 2005 and its main goal and advantage is to allow the exchange of small amounts of data between client and server in order to update asynchronously parts of a web page, without having to recharge it completely.

In a classic web page that AJAX is not used, it would be necessary to refresh the page completely if the content was modified. Thus, AJAX allows web pages more dynamic and provides a faster experience when the user is browsing.

Ajax is used in the development of the project in order to send forms to the server and receive the answer. With this, we can change the database content without the need to refresh the page.

---

<sup>9</sup> References: [18] and [19]

<sup>10</sup> References: [15]

## 2.2. ZURB Foundation<sup>11</sup>

ZURB Foundation is a free framework which is used in order to create web-based software, such as websites and web applications, whose main goal is to develop code faster and better. It uses HTML, CSS, Sass and JavaScript in its templates to develop the applications and it provides templates for typography, forms, buttons, navigation and other interface components, in addition to optional JavaScript extensions.

With regard to ZURB, the makers of Foundation, it is close-knit team of product designers who help companies design and build better websites, services and online products.

I chose ZURB Foundation because all the facilities and all the support that provides in its official website and because it has really useful classes that make the work much easier.

### 2.2.1. A little backstory of Foundation

Foundation had its secret origins in 2008, when the designers of ZURB thought that they needed a framework that allowed them to prototype quickly. Therefore, they decided to take all the global CSS, jQuery plugins, common elements and best practices with the goal of providing a framework that allows programmers build better designed future-friendly sites. The result was the first version of Foundation, which was released in October 2011.

They also realized that mobile was becoming more and more popular — and important — so that they thought they needed a more functional and fluid tool to work on different screens. According to this, in June 2012 they released a major update of Foundation (Foundation 3.0.) that had the advantage to be responsive, being the first open-source framework to be responsive.

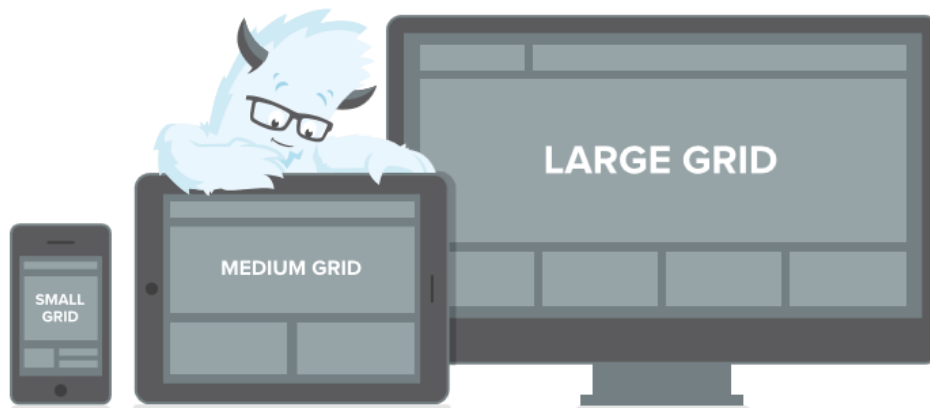
As they also thought that being the fastest way to responsively prototype and build for any device was not enough, throughout the time, ZURB released new updates with improvements and new facilities. For instance, Foundation was the first semantic framework and the first to be Mobile First. In November 2013 ZURB released the last update of the framework, Foundation 5.0.

Nowadays, ZURB is going on investigating and developing new features to ensure Foundation remains the most advanced responsive front-end framework in the world so that the programmers can code smarter and faster.

---

<sup>11</sup> References: [15] and [16]





**Figure 5. Foundation is the first open-source framework to be responsive.**

### 2.2.2. Features

The most important features of the last update of the framework ZURB Foundation are the following:

- Faster for Users

Foundation 5 has a fast core that allows you to create beautifully optimized pages for whatever screen you want. This core also removes unnecessary waiting time for users, especially for animations and transitions. In addition to this, in the last update of Foundation, the style is simpler, cleaner, and faster to restyle.

- Faster to Code

As Foundation is a responsive framework, creating websites and applications for any device is faster than ever. Moreover, there is a new command line tool that allows you to develop Foundation projects very fast.

- Faster to Learn

To help the programmers improve their future-friendly website building skills, the designers of Foundation provide great-quality documentation and a community forum in their official website<sup>12</sup> so that the programmers can learn much more quickly. They have also written a new "Getting Started" guide that can be really useful in the beginning.

---

<sup>12</sup> <http://foundation.zurb.com/>

## 2.3. MySQL Workbench<sup>13</sup>



MySQL Workbench is a unified visual tool for database architects, developers, and DBAs (DataBase Administrators). MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X. Furthermore, it is offered in the following Editions:

- MySQL Workbench Community Edition — Open Source (GPL License)
- MySQL Workbench Standard Edition — Commercial
- MySQL Workbench Enterprise Edition — Commercial

The Standard Edition and the Enterprise Edition have the advantage of offering more facilities than the other edition, but you have to pay for it.

I chose MySQL Workbench to deal with the database because, although it is not required to develop a web based application, it makes the work much easier with the database design and the dealing with the data introduced in this database.

### 2.3.1. A little backstory of MySQL

MySQL was originally founded and developed in Sweden by two Swedish and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's.

MySQL is the world's most popular open source database software because all the advantages that are offered. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for the main companies in the market because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

### 2.3.2. Features

#### *Design*

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler

---

<sup>13</sup> References: [18] and [19]

needs for creating complex ER (Entity-Relationship) models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require a lot of time and effort.

### *Develop*

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, auto-complete and execution history of SQL. The Database Connections Panel enables developers to easily manage database connections. The Object Browser provides instant access to database schema and objects.

### *Administer*

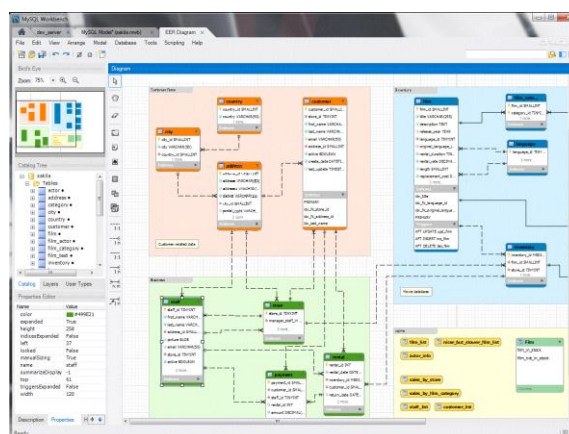
MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases. Developers and DBAs can use the visual tools for configuring servers, administering users, performing backup and recovery, inspecting audit data, and viewing database health.

### *Visual Performance Dashboard*

MySQL Workbench provides a suite of tools to improve the performance of MySQL applications. DBAs can quickly view key performance indicators using the Performance Dashboard. Performance Reports provide easy identification and access to IO hotspots, high cost SQL statements, and more. Moreover, with 1 click, developers can see where to optimize their query with the improved and easy to use Visual Explain Plan.

### *Database Migration*

MySQL Workbench now provides a complete, easy to use solution for migrating Microsoft SQL Server, Sybase ASE, PostgreSQL, and other RDBMS tables, objects and data to MySQL. Developers and DBAs can quickly and easily convert existing applications to run on MySQL both on Windows and other platforms. Migration also supports migrating from earlier versions of MySQL to the latest releases.



**Figure 6.** MySQL Workbench provides an easy to use visual way to design databases

## 2.4. Apache web server<sup>14</sup>



A web server is a computer that stores websites on the Internet and delivers web pages to viewers upon request. This service is referred to as web hosting. Every web server has a unique address, called an Internet Protocol address, that tells other computers connected to the Internet where to find the server on the vast network. Web hosts rent out space on their web servers for people or businesses to set up their own websites, and the web server allocates a unique website address to each website it hosts.

The Apache HTTP server is a HTTP web server of open source for platforms such as Unix (BSD, GNU/Linux...), Microsoft Windows, Macintosh and other, which implements the protocol HTTP/1.12 and the notion of virtual site. The Apache server is developed within the HTTP Server project of the Apache Software Foundation.

Apache is widely accepted in the network: since 1996, Apache is the most widely used HTTP server, reaching its highest market share in 2005 being the server used in 70% of the websites in the world.

Among its advantages we can find the following ones:

- Modular
- Open Source
- Multi-platform
- Extensible
- Popular (easy to get help/support)

The main reason because I chose Apache web server, in addition to the advantages showed above, is because is the server I know best to develop a web based application working with Windows.

---

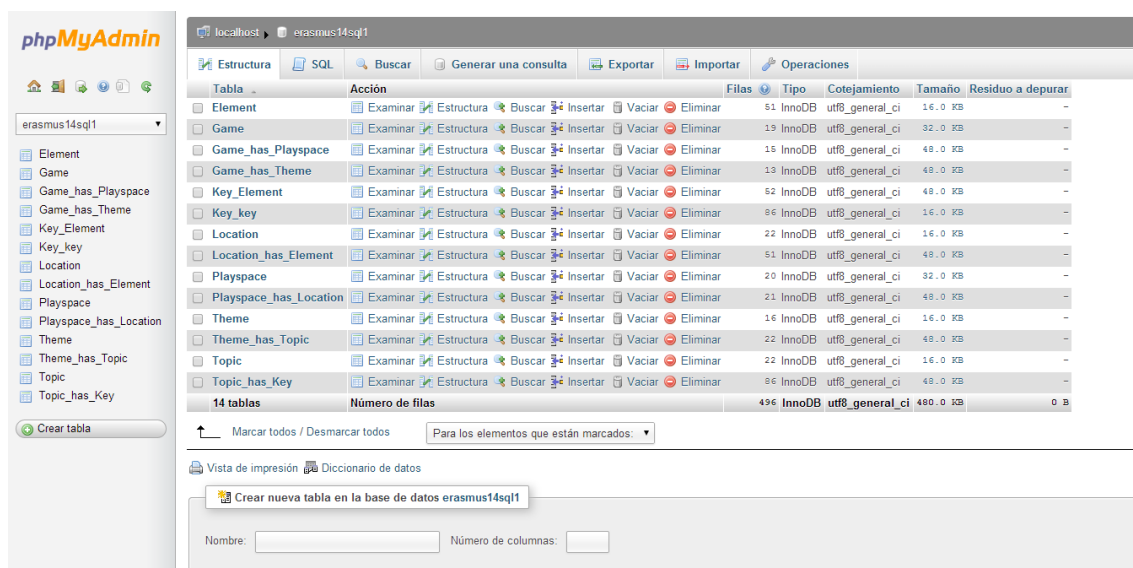
<sup>14</sup> References: [20] and [21]

## 2.5. PhpMyAdmin<sup>15</sup>



PhpMyAdmin is a tool written in PHP intended to handle the administration of MySQL over the Web using Internet tool. Currently it can create and drop databases, create, delete and alter tables, delete, edit and add fields, execute any SQL statement, manage keys on fields, manage privileges, export data into various formats.

During the course of the project it was used in order to upload the database .sql files to the server.



**Figure 7. Screenshot of the database uploaded to the server through phpMyAdmin.**

Main features:

- Intuitive web interface
- Support for most MySQL features: browse and drop databases, tables, views, fields and indexes; create, copy, drop, rename and alter databases, tables, fields and indexes; maintenance server, databases and tables, with proposals on server configuration; execute, edit and bookmark any SQL-statement, even batch-queries; manage MySQL users and privileges; manage stored procedures and triggers
- Import data from CSV and SQL
- Export data to various formats: CSV, SQL, XML, PDF, and so on.
- Administering multiple servers
- Creating PDF graphics of your database layout
- Creating complex queries using Query-by-example (QBE)
- Searching globally in a database or a subset of it
- ...

<sup>15</sup> References: [22]

## 2.6. Conclusion about the tools.

After describing all programming languages and frameworks that I have used during the course of the project, I will explain the main reasons I chose them.

As we have been seeing throughout the chapter, the programming languages chosen for the project were the following:

- PHP (PHP Hypertext Preprocessor).
- HTML (HyperText Markup Language).
- JavaScript.
- CSS (Cascading Style Sheets).
- SQL (Structured Query Language).
- AJAX (Asynchronous JavaScript And XML).

My knowledge about programming was influenced at the moment I decided which languages were necessary to be employed. After working all this time with the selected technologies, I have been able to deepen my learning and achieve a richer knowledge about all of them.

During the development phase of the project I have employed the system infrastructure called WAMP which provides the following tools: Windows as Operating System, Apache as web server, MySQL as database manager and PHP as programming language. Among the most important advantages of WAMP, it is ideal for use as a development server, as well as being easy to install. I have used WAMP as a development tool, since I use Windows as my main Operating System, so I have been testing the web application on the localhost server from my computer, and in the final phase, I have been testing the application in LAMP (Linux, Apache, MySQL and PHP) server.

Regarding the employed frameworks, firstly, I chose ZURB Foundation as a working tool for all features and all the facilities it offers when developing web content. In addition, as mentioned above, on the official website of ZURB Foundation we can find great-quality documentation and a community forum that have helped me a great deal.

Secondly, I used MySQL Workbench C.E. 6.1 for the creation and database management because, besides being a free version, it offers very useful features that make the job much faster and more intuitive, such as the visual way to design databases presented in the previous section.

Finally, I used apache server as the web server and phpMyAdmin to upload the .sql files to the server database, as explained in the previous sections.

## 3. DESIGN

---

The structure in which the chapter is divided is as follows:

Firstly, the most important requirements to be met by the application are described. Secondly, all location-based games discussed in Chapter 1 will be compared (common features and differences). Based on this comparison, I will show the first approach of the design: an abstractions model, trying to be as general and abstract as possible. The final step is to explain the final design along with the user interface design.

### 3.1. Introduction

In order to develop the mobile content editor, I have followed the same methodology that if a web page or web application is developed. Thus, the design of the initial user interface was designed for a desktop computer, taking into account, of course, how the adaptation in the mobile version would be. Therefore, during the course of the project, I have been testing the presentation and functionality of the application especially in the computer where I developed the application (localhost) and checking on a tablet the mobile version worked properly.

### 3.2. Requirements

The requirements that the content editor has to meet are as follows:

#### 3.2.1. Client-Server Architecture

The client-server architecture is a distributed application model in which tasks are distributed among resource or service providers, called servers, and the service requesters, called clients. A client makes requests to another program and the server is the one who gives the answer.

In this architecture, the processing capacity is shared between clients and servers, although the organizational advantages due to the centralization of information management and separation of responsibilities are more important. They facilitate and clarify the system design.

One reason that explains that the application has to follow the client-server architecture is that with the client-server architecture the developer has to build only one database and server-side component and, afterwards, the developer can program different user interfaces according to device type, as in this case: mobile and desktop interfaces.



### 3.2.2. Web based application

A web application consists of the tools that users can use accessing a web server through the Internet or an intranet by means of a browser. In other words, a web application is a software application that is coded in a language supported by web browsers, the ones that run it.

Web applications are popular due to web browser are lightweight clients, the operating system independence as well as the ease to update and maintain web applications without distributing and installing software on potentially thousands of users.

Among the advantages of a web based application I would like to highlight that to work in the web application a person needs only a computer with a web browser and internet connection. Moreover, web applications do not need prior knowledge and a web application has full availability in terms of time and place, people can work on it anytime and anywhere in the world whenever they have internet access.

The application has been developed as a web based application because of the previous advantages that this presents and because of the following reasons. Firstly, the editor could be used in all the different Operating Systems that have installed a browser. In addition to this, being a web based application allows a lot of users to work in parallel and create content in the same database. For instance, two teachers can agree to create content at the same time in the city center of Patras distributing several locations per teacher.

### 3.2.3. Centralized database

A centralized database is a database stored entirely in a single physical location, i.e., it is a storage system that runs on a single computer system without interacting with any other computer in order to keep the data.

The content editor should have a centralized database on the server where the content created by the user will be collected. The content may include descriptions, images, data ... This requirement is a consequence of developing the application with a client-server architecture so that the server centralizes the content of the application.

### 3.2.4. Connection to maps

Since this is an application that must create content for location-based games, it should have some ability or capability to connect with some kind of map, so that the content creator can place the elements he wants on a map to use location identification.

### 3.2.5. Desktop & mobile application

As established in the introductory chapter, it is an application that should run on both desktop computers and mobile devices. The use of ZURB Foundation will facilitate this task, since it is a responsive framework.

Creating content from a desktop computer in this application is sometimes more comfortable for the user, but the mobile application offers another set of interesting benefits. To begin with, for example, if users want to create content for a museum and they don't know which exhibits are exposed in this museum, they can go there with a mobile device and create content in the museum while they are taking a look at the elements there, i.e., at the same time they go around the museum, they are creating content for the game. Another advantage associated to the same scenario is that, while someone is creating new elements in the application, this person will be able to take the photos of the elements with the mobile device camera and, then, upload them to the content editor at the same moment, that is, being in the museum.

### 3.2.6. Different terminology support

Each location-based game has its own physical elements names, its own keys (small sentences) names, and so on. For instance, in MuseumScrabble the objects in the real world that are part of the game are called exhibits while the game is being played. However, in CityScrabble they are called locations or objects. Therefore, we must find a solution to this problem so that, for example, the content creator can choose the names of the physical elements he wants to use when creating or editing content.

## 3.3. Comparison of location-based games

In this paragraph all the common features and the differences among the location-based games presented in the Chapter 1 are detailed. That is, the games are MuseumScrabble, CityScrabble, Benaki Museum Scrabble and Taggling.

The most important feature all the games have in common is the purpose of the game: to achieve all the links (connections) between the physical objects (exhibits, locations, objects and so on) and the keys (hints or digital objects in the example of Benaki Museum Scrabble). All the physical objects belong to a location inside a playspace and the game can have one or more different topics chosen by the team when starting to play. In turn, a topic consists of the keys we mentioned before.

Every relationship has a score which can be 0 or different of 0. Usually, in the games there is only one right relationship, although it can be more than one.

The four games have in their home screens a topic selection. In MuseumScrabble and CityScrabble is the player who has to choose whenever the theme which is going to play. However, in Benaki Museum Scrabble the theme is preset and changes each round. On the other hand, in Taggling we cannot say that there is a topic, because all keys seem to belong to the same topic.

Once the topic is selected, all the apps show:

1. Information about the topic.
2. Clues (keys) to guess the relationship.
3. Button to link them or action buttons to interact with the keys and real objects.
4. Description or relevant information about the QR code scanned.

In some of them you can imagine the progress of the other teams because in MuseumScrabble and in CityScrabble you can see which relationships have been done by the other teams and in Benaki Museum Scrabble when a round finishes, it means that another team have completed all the relationships. On the other hand, in Taggling the player does not know what the score of the rest of the teams is.

CityScrabble and Benaki Museum Scrabble have an intro screen in order to select the session or the kind of game. They also have in every activity a help button.

It also has to be taken into account in the design of the application that there are games that have extra properties that other games does not have, therefore, we need to provide the application of optional properties so that a user that wants to create content for the game CityScrabble can choose the additional properties of CityScrabble, whereas a user that wants to create content for the game Taggling can choose the additional properties of this game. For example, CityScrabble uses two attributes that the other games do not use: a radius of circumference around a real element and a hint of the keys in order to give an idea to the user while playing the game.

Another important difference is the terminology because for each game a different terminology is employed. A solution for this in the design of the application is to create a high level of abstraction, but with this solution the user of the editor will have to understand the terminology chosen for the editor. The best solution, then, is to give to the user the option of choosing the terminology the user wants so that, for instance, if a user wants to create content for MuseumScrabble, he can choose the name hint to call the small sentences that describe the physical elements. In the following table we can see the different terminology for each game and the terminology used during the design of TaggingCreaditor on the right columns of the table:

<b>MuseumScrabble</b>	<b>CityScrabble</b>	<b>BMS</b>	<b>Taggling</b>	<b>Intuitive terminology</b>	<b>General terminology</b>
Museum	Space	Museum	Museum	<b>Playspace</b>	<b>Real object level 1</b>
Rooms, floors...	Squares, street...	Rooms, floors...	Rooms, floors...	<b>Location</b>	<b>Real object level 2</b>
Exhibit	Location or object	Physical archifact	Exhibit	<b>Element</b>	<b>Real object level 3</b>
-	-	-	-	<b>Theme</b>	<b>Virtual object level 1</b>
Topic	Theme	-	-	<b>Topic</b>	<b>Virtual object level 2</b>
Hint	Key	Digital object	Tag	<b>Key</b>	<b>Virtual object level 3</b>

### 3.4. Abstractions: the first approach

Once it is known all the common features and differences of the given location-based games, now, we can move on to design the model that defines abstractions, which should not be too much attached to the specific games, that is, they must be as general as possible.

First of all, if we are going to design an editor to create content for location-based applications, we have the abstraction ‘Game’, so that the user of the editor can create a ‘new game’. In the database of the editor, the user could have more than one created game.

At this point, it is really important to distinguish between the entities that belong to the physical world and exist in the reality and the entities that belong to a virtual world and exist in the mobile phone being content of the game. Now, we will describe the abstractions in both worlds and the relationship between them.

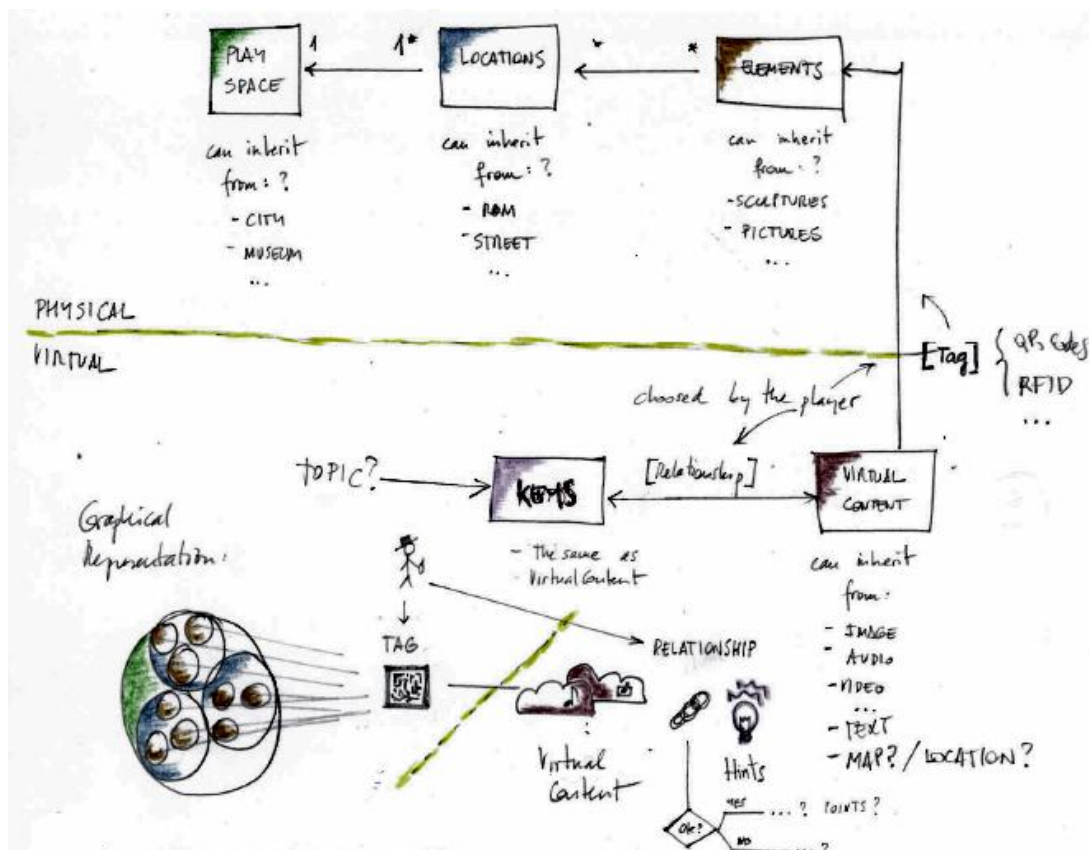


Figure 8. The first sketch of the design which was later modified.

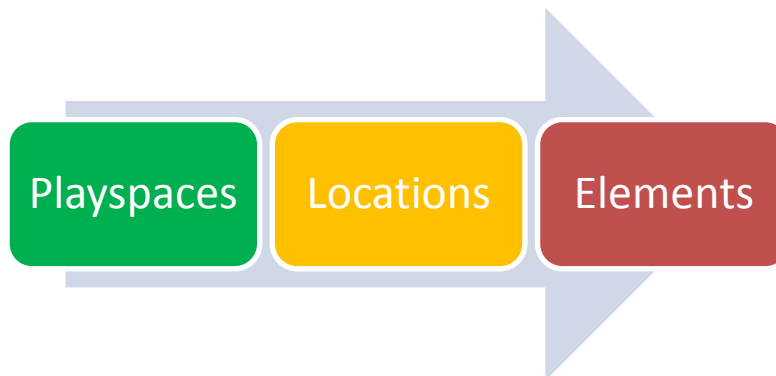
### 3.4.1. Real world abstractions

With regard to the physical world, one of the most important factors to take into account in location-based games is the location where it is played so that these games have a 'playspace'. For example, CityScrabble can be played in multiple locations such as Patras, Athens, or whenever the developer wants if he has content to implement the application. When somebody is developing a location-based game, it is essential to know where the game is going to take place. In the content editor, this will be selected by the user who is creating a new game. Furthermore, the space will have to be prepared for the game. Therefore, 'playspace' will be our first abstraction in the physical world.

In a 'playspace' such as a city center or a museum we can find several locations, i.e., in the city center there are some streets, some squares... whereas in a museum there are several rooms or, even, floors. The abstraction chosen for this will be called 'location', so that a 'playspace' can have one or more 'locations'.

The physical objects are abstractions too and they are the objects with a tag/marker such as a QR code tag, a RFID code tag or something similar. These objects belong to a 'location'. For instance, a physical object can be an exhibit in a museum, a

location in the city center and so on. This will be another abstraction called ‘element’ and is the one in which the tag is placed to prepare a location-based game.



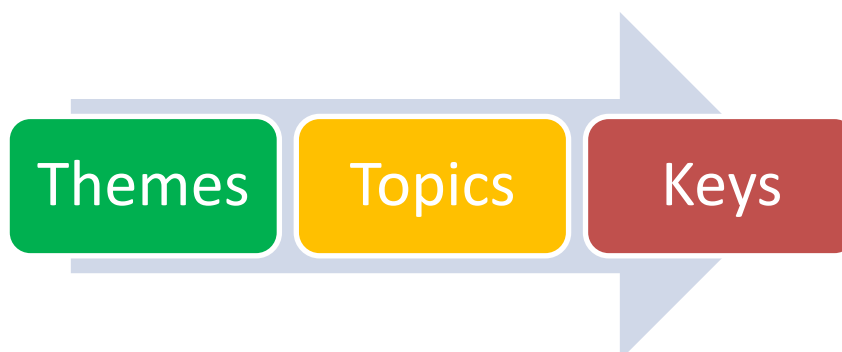
**Figure 9. First approach of the real world abstractions**

### **3.4.2. Virtual world abstractions**

The virtual world, as indicated before, is the one that contains all the content of the location-based game that is included in the devices such as the pictures, the hints and so on.

Each game has different ‘topics’. Therefore, the user of the editor must be able to add new topics. Each topic, in turn, consists of hints or ‘keys’ which are the small sentences that appear in the device of each team while playing. The user of the editor must be able to add new keys to the different created topics.

In addition to this and considering that a list of topics can have some feature in common, we will also have the abstraction ‘theme’. For example, we can have several ‘topics’ whose subject is about some century, then, we can have a ‘theme’ called ‘Centuries’ that contains all these topics.



**Figure 10. First approach of the virtual world abstractions**

### 3.4.3. Relationship between the two worlds

In every game, if a player connects a key with the right physical object the team will add points to their score and, on the other hand, if the key is connected to a wrong tag, the team will not get any points. Therefore, it is clear that the relationship between the two worlds is made by means of these two abstractions: the elements and the keys.

We have to find a proper way to relate the keys and the elements. One first approximation could be by means of scores as it is established in the next example: Imagine a location-based game with three elements and 3 keys with the relationships that are shown in the table below. In this scenario, if one player connects the element 2 with the key 3, 100 points will be added to the score; whereas, if the same player connects the element 2 with the key 2, no points will be added to the score.

	ELEMENT 1	ELEMENT 2	ELEMENT 3
KEY 1	0 points	0 points	100 points
KEY 2	100 points	0 points	0 points
KEY 3	0 points	100 points	0 points

The problem with this way of linking the elements with keys is that it is not a very general way to establish a relationship, it is not flexible. For example, imagine that a user who is creating content for a game does not want to relate an element of the real world with a key through a score, but through a semantic relationship. With the first approach of setting up a score in the relationship, the content creator would not be able to do what he wants. Therefore, we must find a solution to this problem, which will be detailed hereinafter.

### 3.4.4. Summary

In the following table we can see a summary of the first abstraction model. On the left part of the table we can see the abstractions we have to model and on the right part of the table a first approach of the functionalities the content editor should have.

ABSTRACTION	FUNCTIONALITIES THE EDITOR SHOULD HAVE		
Game	New game	Delete game	Edit game
Playspace	New Playspace	Delete Playspace	Edit Playspace
Location	New Location	Delete Location	Edit Location
Elements	New Element	Delete Element	Edit Element
Theme	Add Theme	Delete Theme	Edit Theme
Topic	Add topic	Delete topic	Edit topic
Key	Add key	Delete key	Edit key

### 3.5. Design of the entities hierarchy: final design

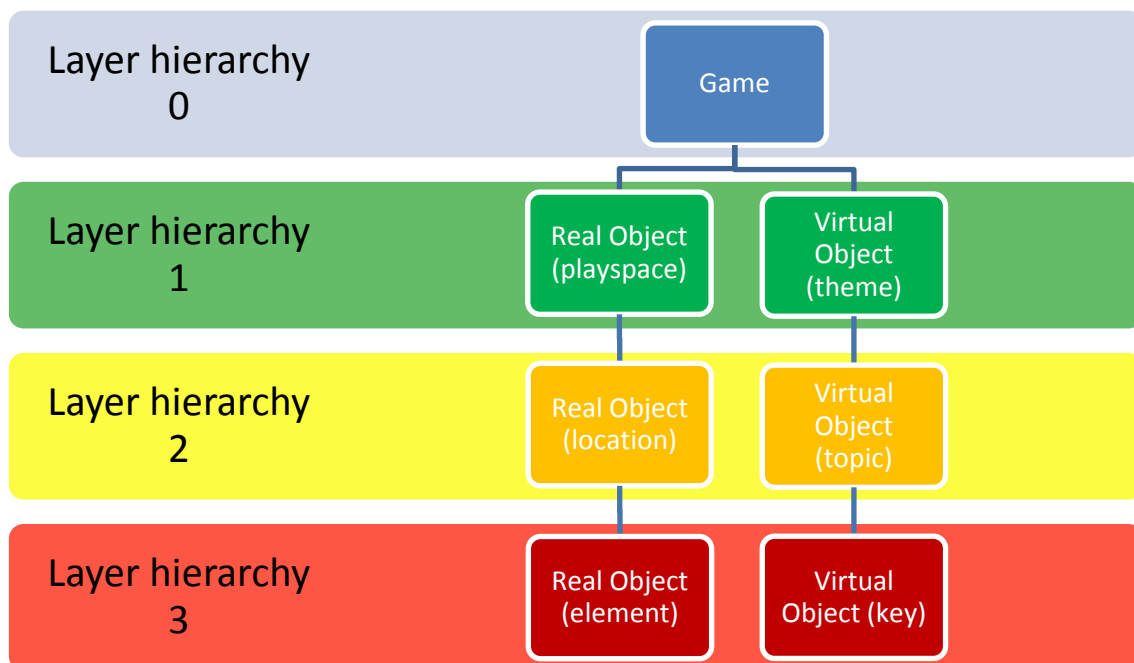
Having reached this point, we realize that the abstraction model that has been proposed is not the best solution since all the names chosen of the abstractions are too specific to the studied location-based games and, therefore, are not general enough. Examples of this are the abstractions ‘topics’ and ‘keys’, which are specific to the scrabble games.

With this problem, the names of the hierarchy are not well suited to all games. For example, in MuseumScrabble instead of elements, objects in the real world, that are part of the game, could be called exhibits, since it is the word used while the game is being played. However, as another example, in CityScrabble, perhaps, the content creator of this game wants to call the physical elements as locations or objects.

As a solution, an entities hierarchy has been designed, in which more general names are used, being a more abstract and flexible model. The design of entities hierarchy that is proposed for the development of the content creator/editor after having studied all the common features of location-based games (see section 3.3) and after having chosen the abstractions (see section 3.4) is the one that can be seen in Figure 11. In it, you can already see the relationship between each of the entities, except for the relationship between the two worlds, which will be detailed during the database design.

One of the goals to be achieved with this solution is to solve the problem described in this point, so that the editor user can choose the names of the entities in the hierarchy. The great advantage is that the editor user, during content creation, does not have to learn or use a different terminology to that used in location-based game he wants to create content.





**Figure 11. Design of the entities hierarchy.**

Inside the boxes of the figure, general names of the entities are shown along with the indicative name that has been used during the course of the project and the design of the database. This indicative name is shown in brackets and is the one I will use from now on to refer to each entity so that, for example, when referring to third-level virtual object, I will use the notation 'key' and when referring to the first-level real object, the notation 'playspace' will be used.

Returning to the design scheme and as I indicated in the previous section, it is important to realize that there is a real world (entities located on the left side of Figure 11) which consists of all real entities such as the exhibits that are exposed in a museum to play the game MuseumScrabble. In addition to the real world, there is a virtual world (entities located on the right side of Figure 11) which consists of all virtual entities such as the keys (small sentences) that are used to play CityScrabble in the center of a city.

The relationship between the real world and the virtual world is carried out between third level entities of both the real and the imaginary world, respectively, i.e., the relationship takes place between the elements and keys. This relationship will be explained in detail in the next paragraph.

In the following each one of the entities will be described in detail, distinguishing between the real world and the virtual world.

### 3.5.1. Game entity

This entity deserves a single paragraph because it is the highest level entity in the general hierarchy. It can be seen as a container of playspaces and themes, taking into account, also, all the attributes of the game. Therefore, a Game could contain many playspaces and many themes (I will explain this with examples in the following paragraphs). This is an abstract entity that belongs to both the real world and the virtual world.

An example to understand the concept of a Game clearer is the following: a 'Game' in the content editor could have the name 'CityScrabble' thus indicating that the user of the editor will create content for that game.

### 3.5.2. Real world entities

#### *Playspace (Layer hierarchy 1)*

It is the highest level entity in the real world and can be considered as a container of locations.

An example of Playspace name, following the example of CityScrabble, could be 'Patras City Center', indicating that the CityScrabble game will be played in the center of Patras.

As I indicated before, one Game can have many playspaces. To put an example: imagine that we want to create content for the CityScrabble for Patras city center and, on the other hand, for Athens city center. By adding to the 'Game' called 'CityScrabble' the playspaces 'Patras City Center' and 'Athens City Center' we will achieve what we wanted.

#### *Location (Layer hierarchy 2)*

It is the entity that follows the playspace entity in the real world hierarchy and can be seen as a container of the physical elements that will belong to content of the location-based game.

An example of Location name, taking the CityScrabble game as example, in the 'Patras City Center' Playspace could be 'George Square'. It means that this square will be part of the game content and that, most probably, will contain several physical elements that will be added in the content of this game.

---

### *Element (Layer hierarchy 3)*

It is the lowest level entity in the entities hierarchy in the real world, but perhaps, the most important. This is because, as mentioned in Chapter 1, one of the main objectives of location-based games is to find the proper relationship between the physical elements which are part of the game with keys (small sentences).

Some examples of Elements, going on with the example of CityScrabble in the playspace 'Patras City Center' and the location 'George Square', could be the 'fountain of George Square' or even 'Caravel shop' that is located in this Square.

### **3.5.3. Virtual world entities**

#### *Theme (Layer hierarchy 1)*

It is the highest level entity in the virtual world and the only role it adopts is the one being a container of topics.

An example of Theme name, following the example of CityScrabble, could be 'Centuries', indicating that the CityScrabble game will have one or more topics related with some century.

As in the case of Playspaces, a Game can have many Themes. To put an example: imagine that we want to create content for the CityScrabble with virtual information (topics and keys) about centuries and, on the other hand, with virtual information about famous people around the world. By adding to the 'Game' called 'CityScrabble' the themes 'Centuries' and 'Famous people' we will achieve what we wanted.

#### *Topic (Layer hierarchy 2)*

It is the entity that follows the Location entity in the virtual world hierarchy and can be seen as a container of the virtual keys that will belong to content of the location-based game.

An example of Topic name, taking as example the CityScrabble, in the 'Centuries' Theme could be 'XVI Century'. It means that the game, most probably, will contain several keys about this century that will have to be linked to the physical elements included in the real world.

### *Key (Layer hierarchy 3)*

It is the lowest level entity in the entities hierarchy in the virtual world, but perhaps the most important. The reason is the same as I indicated for the elements: one of the main objectives of location-based games is to find the correct relationship between the physical elements that are part of the game with keys (small sentences). Therefore, the elements and keys play an important role in creating content for location-based games.

Some examples of Elements, going on with the example of CityScrabble in the playspace 'Patras City Center' and the location 'George Square', could be the 'fountain of George Square' or even 'Caravel shop' that is located in this Square.

## **3.6. User interface design**

The design of user interface or interface engineering is the design of computers, applications, machines, mobile communication devices, software applications and websites focused on user experience and interaction.

It is typically a multidisciplinary activity involving several branches, i.e. the design and knowledge as graphic design, industrial, web, software ... Its main goal is to get applications or objects more attractive and also make user interaction as intuitive as possible. In short, follow the user-centered design.

The graphical user interface (GUI) is a computer program that acts as user interface, using a set of images and graphic objects to represent the information and actions available in the interface. Its main use is to provide a simple visual environment to allow communication with the developed system.

Usually, programmers tend to follow the model of the programmer to design the user interface, which does not take into account the user's way of thinking or the user's software perspective. The user interface of TaggingCreaditor has been designed in order to be as intuitive as possible and trying, at all times, taking into account the user's mindset and perspective of the computer screen or of a mobile device.

To do this, we should be aware of the cognitive and perceptual abilities of individuals to adapt the application to them, i.e., avoid the user having to use their short-term memory.

As indicated in previous chapters of the document, to design the user interface it must be considered that the application should run on both desktop and mobile.

For this reason, in the first design was taken into account this requirement, making a design for desktop computer that was easily adaptable to a tablet screen or a smaller mobile device. Thus, if the design of the user interface of the desktop is perfectly adaptable to the user interface of mobile device, we will not have to make any changes

to files or to adapt the code to the mobile version. However, this is an ideal situation and, therefore, files for the user interface of the desktop version and files for the user interface of the mobile version will have to be programmed.

The files in which the user interface of the mobile interface will be programmed will have as a starting point the desktop version developed code, and from there, appropriate modifications will have to be made in order for the mobile user experience to be as more satisfying and intuitive as possible.

Furthermore, another fundamental reason for having followed this way of working is that the development of the code is going to be programmed on a computer screen, which will greatly facilitate testing during the programming phase of the project and will not have to be tested with a mobile device, what would make the development slower.

For the design of the user interface was taken into account that three main screens would be needed:

-Real objects screen: In it playspaces, locations and elements will be managed. In Figure 12 can be seen the sketch of the initial design of this screen that was created during the design phase of the user interface.

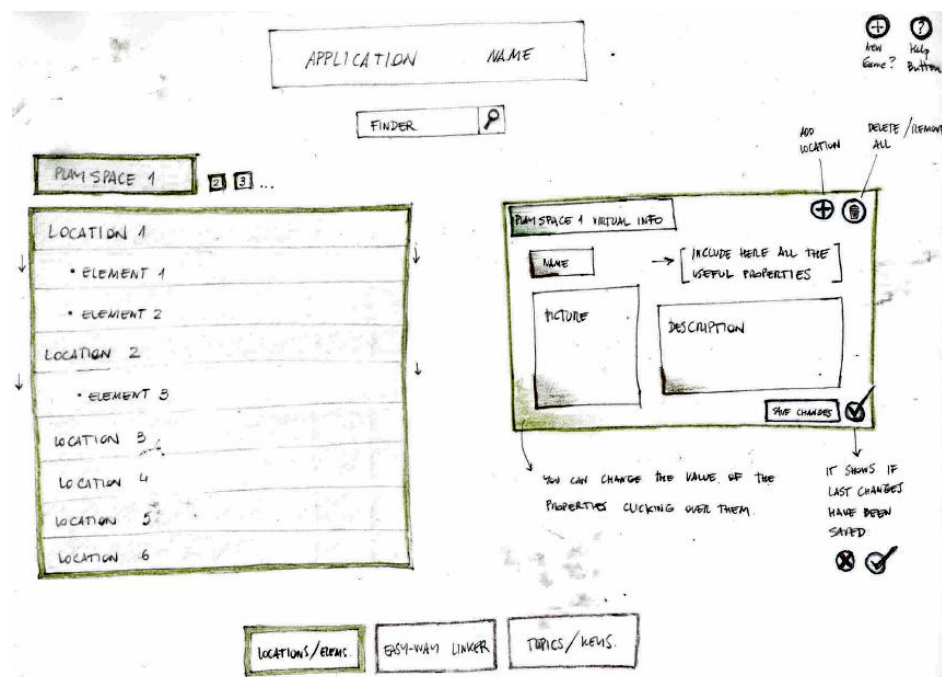


Figure 12. Sketch of the design of the real objects management screen.

-Virtual objects screen: In it, themes, topics and keys will be managed. Figure 13 shows a sketch with the initial design of this screen.

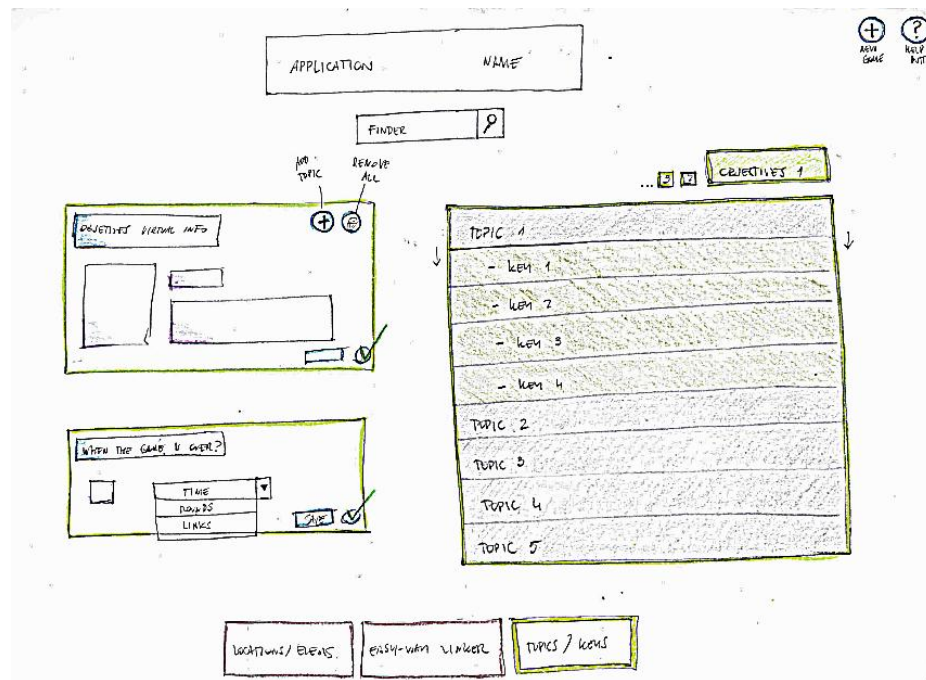


Figure 13. Sketch of the design of the virtual objects management screen.

- Screen of relationship between real and virtual objects: The main mission to be met in this screen is to relate the elements with keys. Figure 14 shows a sketch with the initial design of this screen.

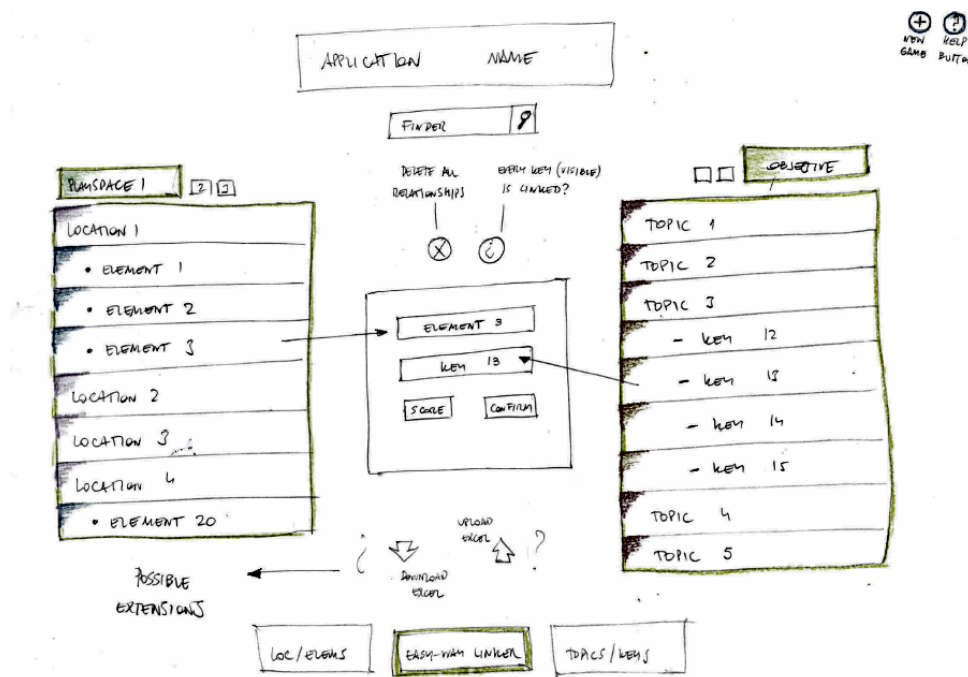


Figure 14. Sketch of the design of the linker screen.

It is important to remark that during the design of the user interface it has been trying to organize the elements of the user interface in columns clearly differentiated, so that when a mobile device is used to create content in the editor, these columns are located in a single column, ordered online. With this in mind, the number of changes needed to adapt the desktop to the mobile version will be much lower.

In conclusion, although it is obvious, this design of the user interface has been followed by a flexible way, i.e. according to current needs some components of the interface have been adding or deleting components. Therefore, during the programming phase, we realized about the need to add a fourth screen in which a list of created games appears. The design of this screen followed the same philosophy as others: divide the interface into columns so that it can be easily adapted, afterwards, to a screen of a mobile device.





## 4. APPLICATION - DEVELOPMENT AND IMPLEMENTATION

---

In this chapter I will present the core of the work I have done. In it, I will describe in more detail the implementation, development and implementation beginning, firstly, with an introduction where I comment how the project was planned, its structure and other relevant information. Furthermore, I will explain the design of the database, the implementation of end-user interface, the main functionalities of the application and, finally, I will describe an scenario to which a school teacher must create content.

## 4.1. General aspects

The project started in February 2014 and was completed in July 2014 and has gone through several phases of development in a pattern very similar to the model of software engineering called waterfall model, although it has not been strictly followed. This model is the methodological approach that orders strictly the process steps for developing software in such a way that the beginning of each phase must wait for the completion of the previous phase. Thus, any design error detected in the testing phase necessarily leads to redesign and reprogram of the relevant code, increasing development costs. Among its advantages I can say that it is an easy model to implement and understand that promotes an effective working methodology: define before design, design before coding.

### 4.1.1. Phases and organization of the project

The phases that were followed during the project are:

1. Phase of initial study: Study of the general aspects of location-based games, along with further study of several proposed games at the beginning trying to compare them looking for common features and differences.
2. Phase of requirements analysis: Study of the requirements that the application should meet.
3. Design phase: Starting with the design of entities that are the more abstract possible and ending with the design of the user interface.
4. Programming phase: This phase consists of several sub-phases such as the programming a prototype user interface, the programming of the functionalities and the programming of the final user interface.
5. Testing and debugging phase: The last days on the project have been spent in testing the application performing simulations of existing content insertion of several location-based games. When an operation error was detected, I solved it.

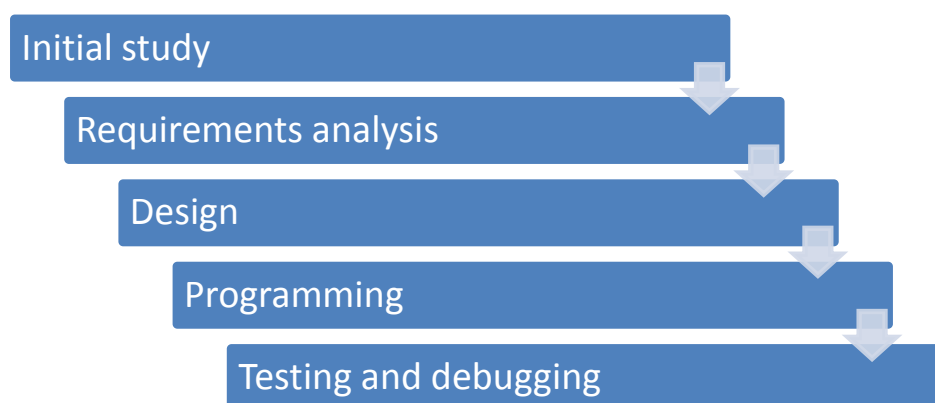


Figure 15. Phases in which the project has been developed.

Regarding the phases described, the most complicated of the project was to find the more possible abstract entities during the design phase. Most of the time has been dedicated to study well the operation of all the proposed games and create designs that could be as abstract as possible. The other great part of the time has been spent to program the application.

For the project development I used the technologies described in Chapter 2. Firstly, for the main functionalities I used HTML (embedded in PHP), PHP and JavaScript. It should also be noted that I have included JavaScript libraries such as JQuery along with some useful plug-ins such as datatables (table plug-in for jQuery). On the other hand, about the style, CSS code has been used. To write this code I have tried, in order to be clear, to apply it to the id HTML attribute of the labels, but sometimes I have needed the class HTML attribute. Finally, in order to interact with the database I have used MySQL code and a database divided in 14 tables trying to be the most flexible and well-working possible.

Since the beginning of the programming phase, I tried to organize the code in the clearest possible manner following, as far as possible, the programming paradigm called modular programming, in order to make the application the more readable and manageable possible.

To accomplish this, I have separated the PHP code into 51 files, each implementing a different task and two of them, dedicated exclusively to the support of the English and Spanish languages. In addition to this, all the JavaScript code (including jQuery) and CSS has been, mostly, included in separated external files.

Among all these files, four of them were aimed to the mobile graphical user interface. Each was dedicated to each of the main application screens: game list screen, real world management screen, virtual world management screen and linker screen. The starting point of these files was the desktop files used for the same purpose. It was required to make several adaptations in order for the mobile users to have a better experience while using TagginCreaditor.

### 4.1.2. Differentiation between mobile and desktop

In the Figure 16, we can see the way it is distinguished in the code between a mobile device and a desktop one in the first screen of the application. As we can see, it is a script in javascript code in which we use the *screen.width* attribute to decide about it.

```
<script TYPE="text/javascript">
function gameList(){
    if(screen.width<1024){
        location.href="/php/gameList_mobile.php";
    }else{
        location.href="/php/gameList.php";
    }
}
function newGame(){
    if(screen.width<1024){
        location.href="/php/newGame.php?mobile=m";
    }else{
        location.href="/php/newGame.php";
    }
}
</script>
```

Figure 16. JavaScript script which allows to recognize a mobile device.

In the rest of the files the sessions are used so that it is not required to program script that redirect to other pages like in the previous example. In the next figure it can be seen the first lines of code of *gamelist\_mobile.php* file, which is responsible for showing the list of the created games and a details box that shows the details of a game.

```
<?php
include('initialize.php');
$_SESSION['mobile'] = "m";
if (!$_SESSION['lang']){
    $_SESSION['lang'] = "en";
}
include('changeLang.php');
include('lang/'.$_SESSION['lang'].'.php');
?>
```

Figure 17. First lines of *gamelist\_mobile.php* showing the use of sessions.

Regarding the Figure 17, the file called *initialize.php* is in charge of initializing the sessions. It can also be seen that, in addition to the use of sessions when differentiating the mobile version of the desktop version, sessions are also used to change the language of the user interface, so that the application supports English and Spanish.

## 4.2. Database design

One of the last steps before starting to develop the user interface with all the features of the application is the design of the database. In the Figure 18 we can see a sketch of the database design that was created during the course of the project whereas in the Figure 19 we can see the final database design for the content editor, which was created using MySQL Workbench 6.1 CE.

As we can see in the sketch, the first database design had a few attributes in the tables in addition to have some differences, compared to the final design. For instance, one noticeable difference is in the 'objective' table, which is called theme in the final design.

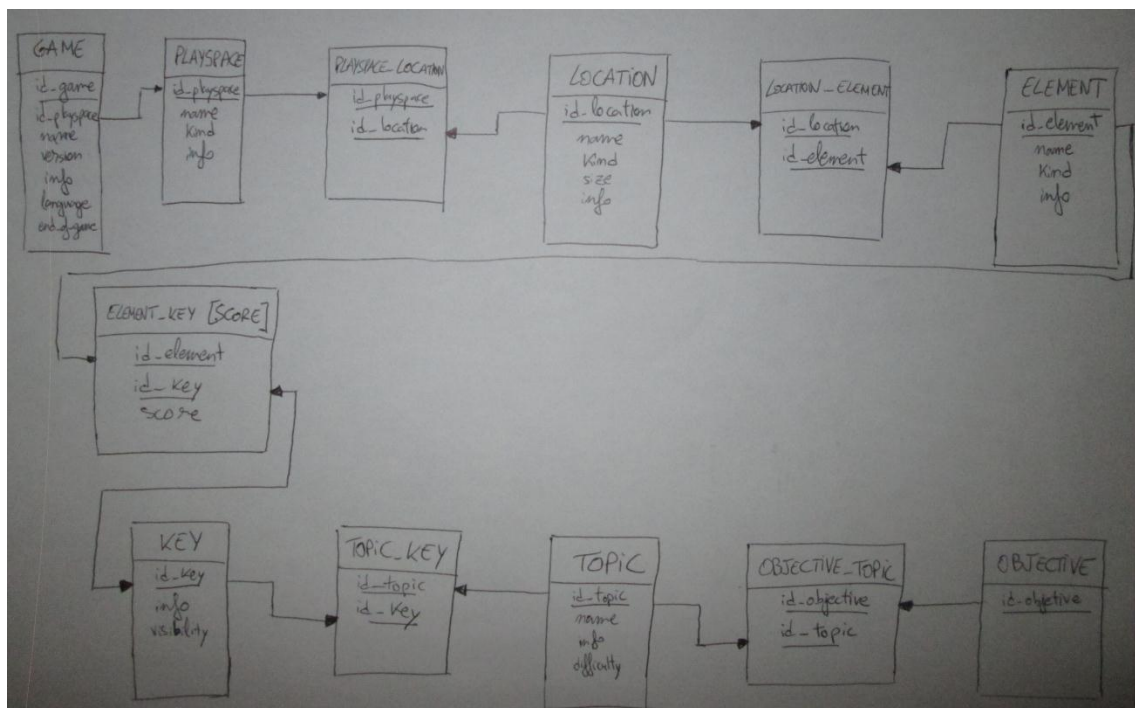


Figure 18. Sketch of the database design created before the final design.

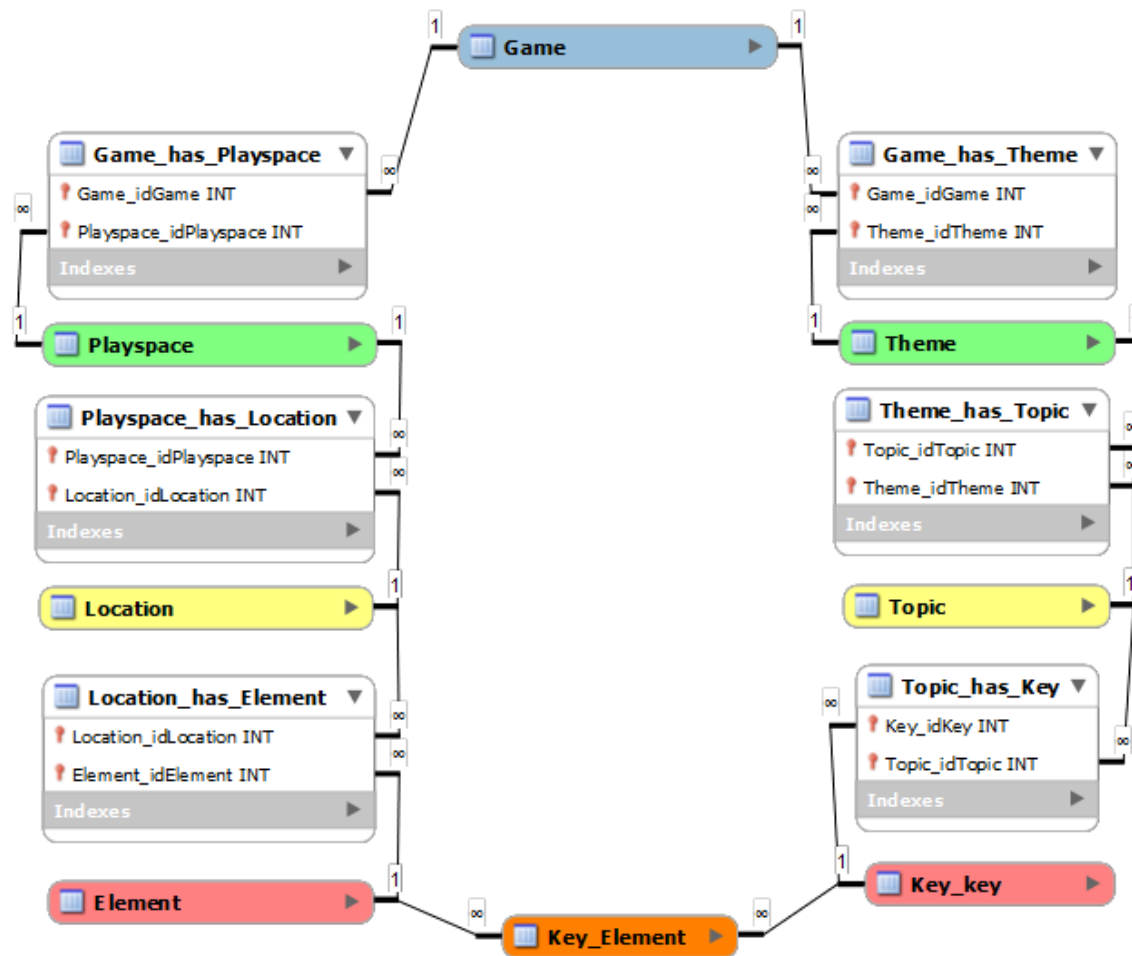


Figure 19. Database design in MySQL Workbench 6.1 CE.

At first sight we can see that the red entities are the ones of the third level of the entities hierarchy (elements and keys), the yellow entities are the entities of the second level of the entities hierarchy (locations and topics), the green entities are the ones of the first level of the entities hierarchy (playspaces and themes) and, finally, that the blue entity is the one of the highest level of the hierarchy (Game).

The other tables of the design, in white, are tables of relationship between entities. The orange table is also a relationship table, but it is of a particular importance because it establishes the relationship between the real world and the virtual world. The types of relationships between entities in the database design are:

- Relationship Game – Playspace & Game – Theme: As I have been discussing throughout the paper, a game can contain many playspaces. As we want the content editor as flexible as possible, also, a playspace may belong to different games. For example, the playspace 'Patras City Center' may be included in the game 'CityScrabble' and the game 'Taggling'. This property makes the relationship between the entities and playspace game is many-to-many. The

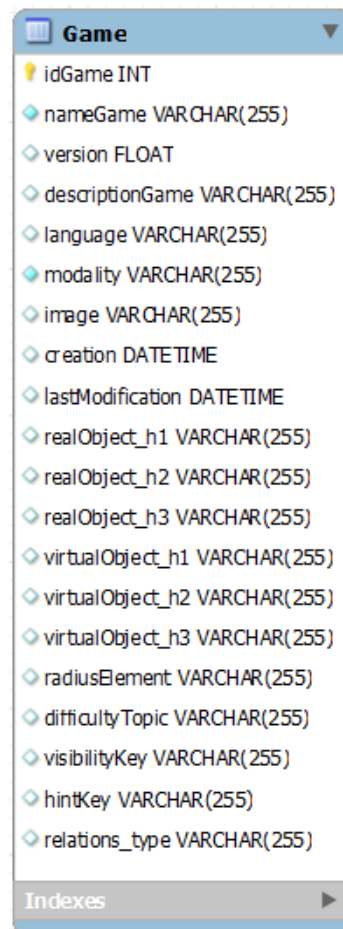
same explanation can be adopted to the relationship between game and themes, since a game can have many themes and theme can belong to several games.

- Relationship Playspace – Location & Theme – Topic: The relationship between level 1 and 2 of the entities hierarchy is considered too many-to-many because a playspace may contain many locations and location can belong to several playspaces. The same applies to the relationship between themes and topics.
- Relationship Location – Element & Topic – Key: As in the relationship between level 1 and 2 of the entities hierarchy, the same happens in the relationship between level 2 and 3. This is a many-to-many relationship because a location may contain many elements and an element can belong to several locations. The same applies to the relationship between topics and keys.
- Relationship Element – Key: This is the most special relationship of the design of the database because, as discussed below, includes the essential attributes to relate the elements and keys. As the above cases, this relationship is also many-to-many because an element may relate to different keys, and a key may be associated with several elements.

As has been seen, all relationships are many-to-many relationships with the aim of achieving an editor as flexible as possible. In the case of many-to-many relationships is needed a relationship table for each of the relationships. In the Figure 19 you can see how for each relationship there is a white table between the entities, except for the special case of the relationship between the elements and keys, which is orange.

In the following paragraphs I will focus on the detailed description of the attributes that are included in the database of each design entities.

### 4.2.1. Game attributes



**Figure 20. Game entity in the database (MySQL Workbench 6.1 CE)**

-idGame: Number (INT) used to identify each game. It is chosen automatically by the database.

-nameGame: Character string (VARCHAR-255) with the name of the game.

-version: Number (FLOAT) in which we can enter the version of the game. We can have the same game with two different versions.

-descriptionGame: Character string (VARCHAR-255) with the description of the game.

-language: Character string (VARCHAR-255) indicating the language of the game. Possible options: English, Greek or Spanish.

-modality: Character string (VARCHAR-255) indicating the manner in which a game ends. A game can finish because of three reasons:

-Time: Establishing a limitation of time.

-Rounds: Establishing the game has to finish after a number of rounds.



- Links: We can also decide that the game doesn't have to finish until all the relationships between the elements and keys are carried out.
- image: Character string (VARCHAR-255) with the path of the game image stored on the server. This image could be the logo of the game.
- creation: Date (DATETIME) indicating when the game was created.
- lastModification: Date (DATETIME) of the last modification in the attributes of the game
- realObject\_h1, realObject\_h2, realObject\_h3: Character strings (VARCHAR-255) that name real objects of first (h1), second (h2) and third (h3) level of the entities hierarchy, respectively. The user, when creating a new game, has the possibility of changing the default values 'Playspace', 'Location' and 'Element' for the values the user prefer, which gives to the editor a great flexibility (you can also modify these settings in the edition of the game). These attributes are really useful because the content editor user will be able to choose the names of the entities in the hierarchy so that, for example, if someone wants to create content for CityScrabble, the realObject\_h3 string would be 'objects', whereas, if someone wants to create content for MuseumScrabble, the realObject\_h3 string would be 'exhibits'.
- virtualObject\_h1, virtualObject\_h2, virtualObject\_h3: Character strings (VARCHAR-255) that name virtual objects of first (h1), second (h2) and third (h3) level of the entities hierarchy, respectively. As with the names of real objects when creating a new game (or editing an already created), the content editor user can modify the names of entities in the virtual world at any time.
- radiusElement: This attribute is part of the optional properties when the game is being created or when its main properties are being modified. The reason for its existence in the database is that it is necessary for the content creation of games such as CityScrabble game, since the physical elements must designate the radius is required on the location map. This game attribute indicates whether the radius attribute in the element entity (discussed below) makes sense in the elements of a game or no sense. If the content creator decides not to add this optional feature for a particular game (value 'NoRadiusSet' in radiusElement), the elements of that game will not have this feature (it will be hidden in the user interface). On the other hand, if the attribute makes sense, the database will contain the default value of the radius of the elements.
- difficultyTopic: This attribute is part of the optional properties when the game is being created or when its main properties are being modified. The reason for its existence in the database is that the content creator can decide whether assigning difficulty to a topic or not assigning it. Therefore, this game attribute indicates if the difficulty

attribute of the topic entity (discussed below) makes sense or it does not. If the content creator decides not to add this optional feature for a particular game (value 'NoDifficultySet' in difficultyTopic), the topics of that game will not have this feature (it will be hidden in the user interface). On the other hand, if the attribute makes sense, the database will contain the default value of the difficulty of topics.

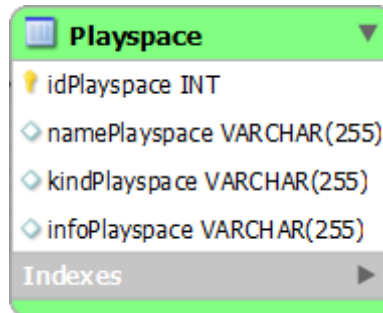
-visibilityKey: This attribute, as the previous ones described above, is part of the optional properties when the game is being created or when its main properties are being modified. The reason for its existence in the database is that it is needed for gameplay of games such as Taggling, since, as I explained in the section 1.3.4, some keys could be placed in exhibits and other hidden on devices (the visibility attribute of the key entity indicates, then, where the key will be located). Therefore, this game attribute indicates whether the visibility attribute of the key entity makes sense or it does not. If the content creator decides not to add this optional feature for a particular game (value 'NoVisibilitySet' in visibilityKey), the keys of that game will not have this feature (it will be hidden in the user interface). On the other hand, if the attribute makes sense, the database will contain the default visibility of the keys.

-hintKey: This attribute is another one being in the group of the optional properties when the game is being created or when its main properties are being modified. The reason for its existence in the database is that it is necessary for the content creation for games such as CityScrabble game, since the keys must designate a hint. Therefore, this attribute indicates whether the hint attribute in the key entity makes sense or it does not. If the content creator decides not to add this optional feature for a particular game (value 'NoHintSet' in hintKey), the keys of that game will not have this feature (it will be hidden in the user interface). On the other hand, if the attribute makes sense, the database will contain the default value of the hint of the keys.

-relations\_type: This attribute also belongs to the group of the optional properties when the game is being created or when its main properties are being modified. The reason for its existence in the database is that each location-based game can have different types of relationships: through scores, through semantic sentences, and so on. Therefore, if the content creator decide to add this optional feature, this game attribute indicates the number of relationship types to be able to choose when linking elements with keys and the names of such relationships (all in a string separated by '//'). For example, if we want the elements and the keys of a game can be related through the types 'Score', 'Sentence', 'Default', the relations\_type attribute will contain the following string in the database: '3//Score//Sentence//Default'. If the content creator decides not to add this optional feature for a particular game (value 'NoTypeSet' in relations\_type), the user of that game will not be able to choose any type of relationship between the elements and keys (it will be hidden in the user interface).

### 4.2.2. Attributes of the real world entities

#### *Playspace*



**Figure 21. Playspace entity in the database (MySQL Workbench 6.1 CE)**

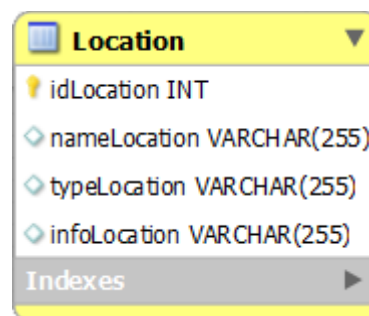
-idPlayspace: Number (INT) that identifies each playspace introduced into the editor. It is chosen automatically by the database.

-namePlayspace: Character string (VARCHAR-255) with the name of the Playspace.

-kindPlayspace: Character string (VARCHAR-255) with the type of Playspace. It will be a free text box so that the user can choose whatever he wants.

-infoPlayspace: Character string (VARCHAR-255) with information about the Playspace.

#### *Location*



**Figure 22. Location entity in the database (MySQL Workbench 6.1 CE)**

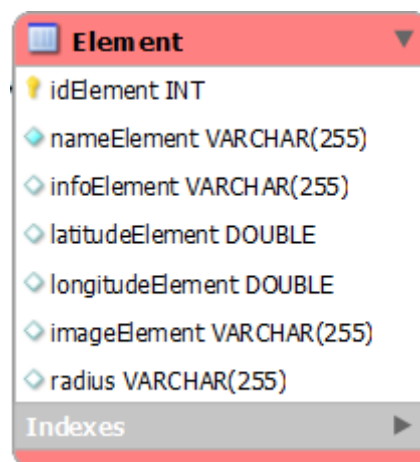
-idLocation: Number (INT) that identifies each location introduced into the editor. It is chosen automatically by the database.

-nameLocation: Character string (VARCHAR-255) with the name of the Location.

-typeLocation: Character string (VARCHAR-255) with the type of Location. It will be a free text box so that the user can choose whatever he wants.

-infoLocation: Character string (VARCHAR-255) with information about the Location.

## Element



**Figure 23. Element entity in the database (MySQL Workbench 6.1 CE)**

-idElement: Number (INT) that identifies each element introduced into the editor. It is chosen automatically by the database.

-nameElement: Character string (VARCHAR-255) with the name of the element.

-latitudeElement: Number (DOUBLE) indicating the latitude of the location of the element. It is used to place the element on the application map.

-longitudeElement: Number (DOUBLE) indicating the longitude of the location of the element. As the latitude, it is used to place the element on the application map.

-imageElement: Character string (VARCHAR-255) with the path of the element image stored on the server. This picture could be a picture of the element.

-radius: Character string (VARCHAR-255) which indicates which is the radius taken by the element in the location map. As I explained formerly, this attribute is optional and the reason for its existence in the database is that it is necessary for the content creation for the CityScrabble game, since the physical elements must designate the radius is required on the location map. If the content creator did not include this feature previously to the game that the element belongs, the user will not have the option to enter a value for this attribute.

### 4.2.3. Attributes of the virtual world entities

#### *Theme*

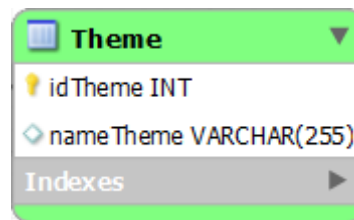


Figure 24. Theme entity in the database (MySQL Workbench 6.1 CE)

-idTheme: Number (INT) that identifies each theme introduced in the application.

-nameTheme: Character string (VARCHAR-255) with the name of the theme.

#### *Topic*

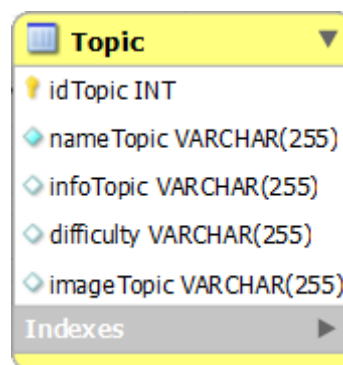


Figure 25. Topic entity in the database (MySQL Workbench 6.1 CE)

-idTopic: Number (INT) that identifies each topic introduced in the application.

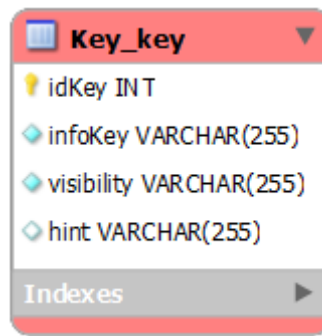
-nameTopic: Character string (VARCHAR-255) with the name of the topic.

-infoTopic: Character string (VARCHAR-255) that provides information about the topic.

-difficulty: Character string (VARCHAR-255) that serves as an indication of the difficulty of relating the keys in this topic to the proper elements. As mentioned in the game attributes, this attribute is optional, so if the user of the editor did not include it previously for the game that the topic belongs, the user will not have the option to enter a value for this attribute.

-imageTopic: Character string (VARCHAR-255) with the path of the topic image stored on the server. This picture could be a significant picture about the topic.

## Key



**Figure 26. Key entity in the database (MySQL Workbench 6.1 CE)**

-idKey: Number (INT) that identifies each key introduced in the application.

-infoKey: Character string (VARCHAR-255) will be the small sentence that contains information about some element or elements. During the gameplay of the location-based games, players have to relate these small sentences with one or more physical elements properly.

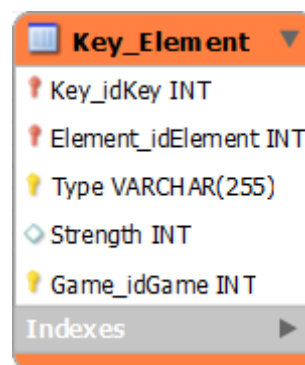
-Visibility: Character string (VARCHAR-255) used to indicate where the key must be placed. As mentioned in the game attributes, this attribute is optional and the reason for its existence in the database is that it is necessary for the content creation for Taggling game, since some keys can be placed in exhibits and other hidden devices of the players. If the content creator did not include this feature previously to the game that the key belongs, the user will not have the option to enter a value for this attribute.

-hint: Character string (VARCHAR-255) used as a hint or tip in the game. As with the visibility attribute, this attribute is optional and the reason for its existence in the database is that it is necessary for the content creation for MuseumScrabble game, since the keys must designate a hint. If the content creator did not include this feature previously to the game that the key belongs, the user will not have the option to enter a value for this attribute.

#### 4.2.4. Relationship Element-Key

As has been seen throughout this chapter, the relationship between the real world and the virtual world plays a very important role to create content in location-based games. The main reason is that all the studied games aim to relate the keys provided by the mobile application with real-world objects that are part of the game. Therefore, it is clear that the relationship between the two worlds is made by means of these two entities: the elements and the keys.

As discussed in Section 3.4.3, a first approach when relating the elements with keys is through a score. The problem with this is that it is not a very flexible solution. To fix this, it was decided to have two attributes (type and strength) that make the relationship is more flexible. In what follows, I will present all the attributes that characterize the Key-Element relationship.



**Figure 27. Relationship element-key entity in the database (MySQL Workbench 6.1 CE)**

- Key\_idKey: Number (INT) of identification of the key being part of the relationship.
- Element\_idElement: Number (INT) of identification of the element being part of the relationship.
- Type: Character string (VARCHAR-255) indicating what type of relationship there is between the element and the key. The relationship types are the ones that the user indicated in the game creation or edition, i.e., in the relations\_type attribute of the game entity.
- Strength: Number (INT) with value in the range 0 to 100 that indicates the strength of relationship between the selected element and key. If a key and an element are closely related, this value will be high (closer to 100), whereas, if a key and an element are slightly related, this value will be lower (closer to 0).
- Game\_idGame: Number (INT) of identification of the game that this relationship belongs.

### 4.3. User interface implementation

The main screens which are part of the application are: Welcome screen, game list screen, real world management screen, virtual world management screen and the linker screen. In the following I am going to describe all of them and, afterwards, I will show how to navigate through them in the application.

#### 4.3.1. Main screens

##### *Welcome screen*

This screen simply welcomes the user of the application and gives the options 'Select Game' or 'Create Game'. In the first option the user can check or edit the content data of a previously created game and the second option will allow the user to create content for a new game.



Figure 28. Welcome screen of TaggingCreaditor.



## Game list screen

This screen is divided into two main parts:

- List of games: This part of the screen the list of games that the user created is displayed. For the game list was used JQuery plug-in called 'datatables'.
- Game Details: Every time the user chooses a different game in the game list, the part of game details will show the main features of the selected game.

From this screen, in addition to have access to the edition of a created game or to the creation of a new game, we can perform other secondary functions such as editing main features of the game that appear in the part of game details or the access to the editing screen of all properties of the game.

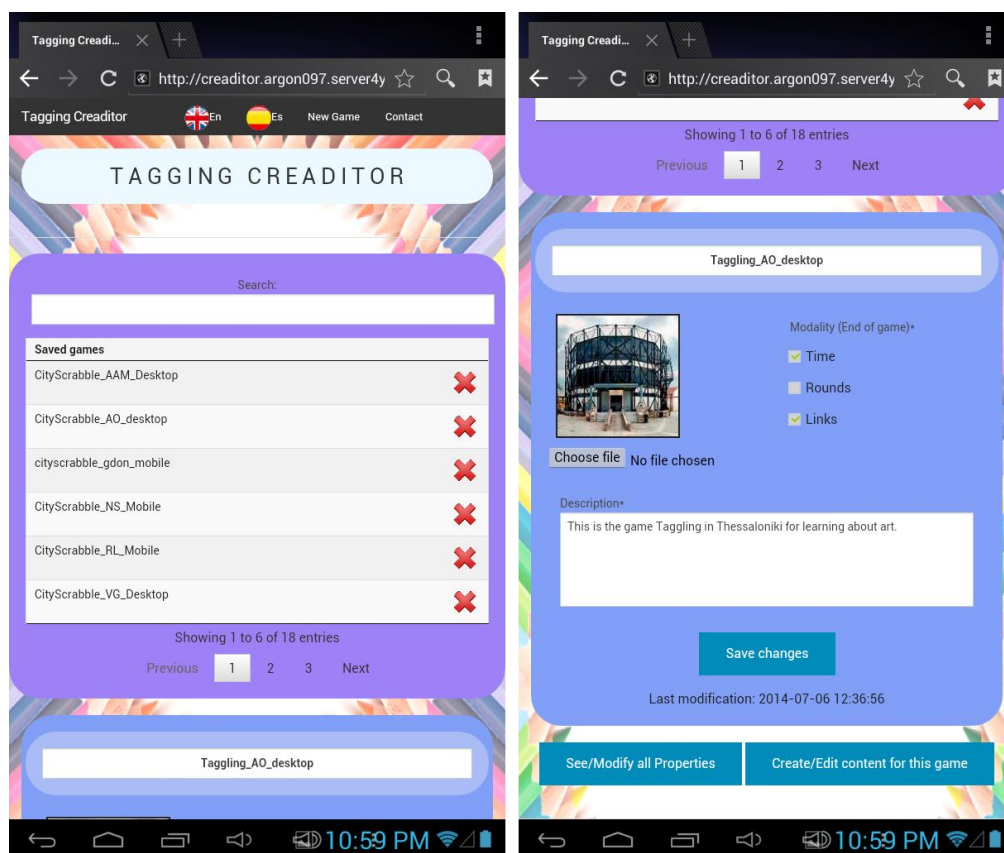


Figure 29. Game list screen of TaggingCreaditor (left: games list, right: game details)

### Real world management screen

Once we access to the creation (or edition) of the content of a created game, the user will see this main screen. Here, the user can manage (create, modify features or remove) all physical entities that will be part of the corresponding location-based game, i.e., the user can create physical objects of first, second and third level of the entities hierarchy (playspaces, locations and elements in the intuitive terminology, respectively).

This screen is divided into two main areas:

-‘Navigation List’ area: The user can navigate through this navigation list to see all the physical objects that have been included in the game content.

-‘Entity Details’ area: Each time the user selects a physical entity in the navigation list, this detail area shows the assigned content of the main features of the physical entity selected.

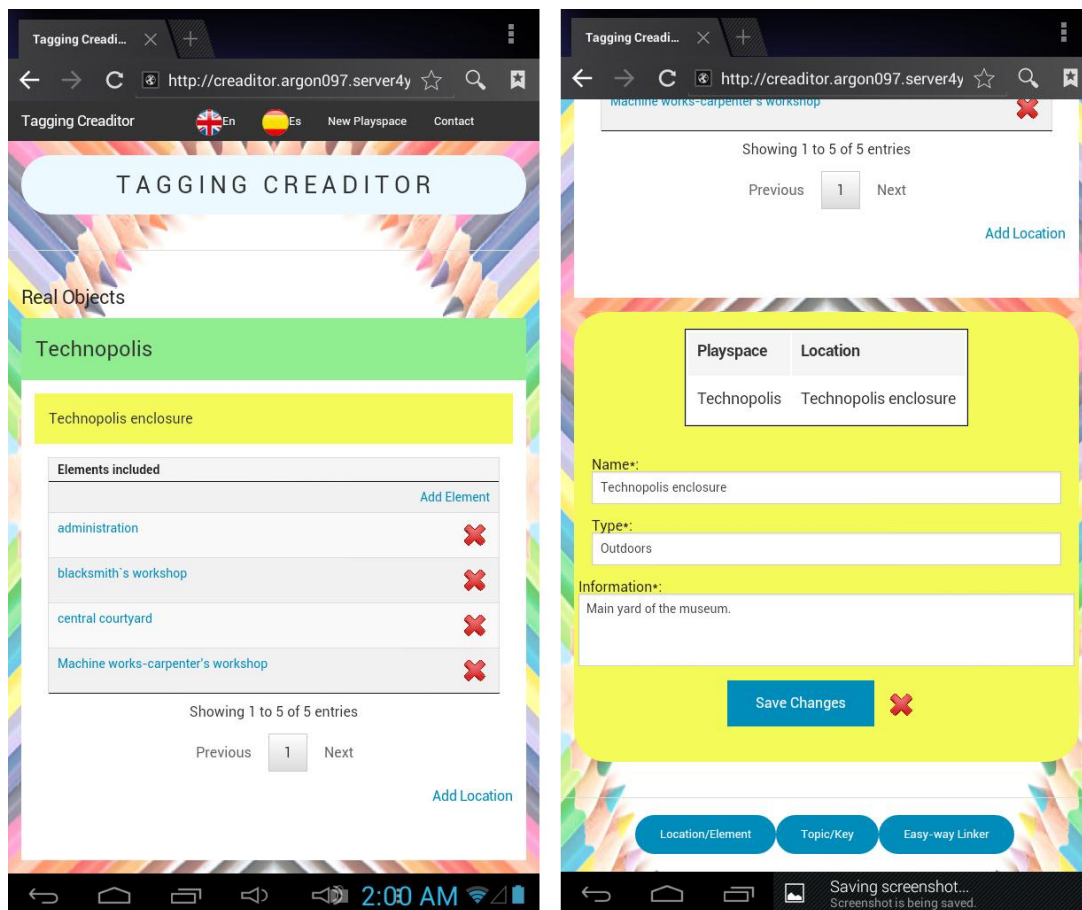


Figure 30. Real world management screen (left: navigation list, right: entity details boxes)

### Virtual world management screen

Going on with the creation (or edition) of content for a game, it is also required to create all virtual entities that will be part of the game, i.e., the creation of virtual objects of first, second and third level of the entities hierarchy (themes, topics and keys on the intuitive terminology, respectively). In this screen the user has the possibility to manage these virtual entities: create them, modify them or delete them.

As was the case in the screen of creation/edition of physical entities, this screen is divided into two main areas:

- ‘Navigation List’ area: The user can navigate through this navigation list to see all the virtual objects that have been included in the game content.
- ‘Entity Details’ area: Each time the user selects a virtual entity in the navigation list, this detail area shows the assigned content of the main features of the virtual entity selected.

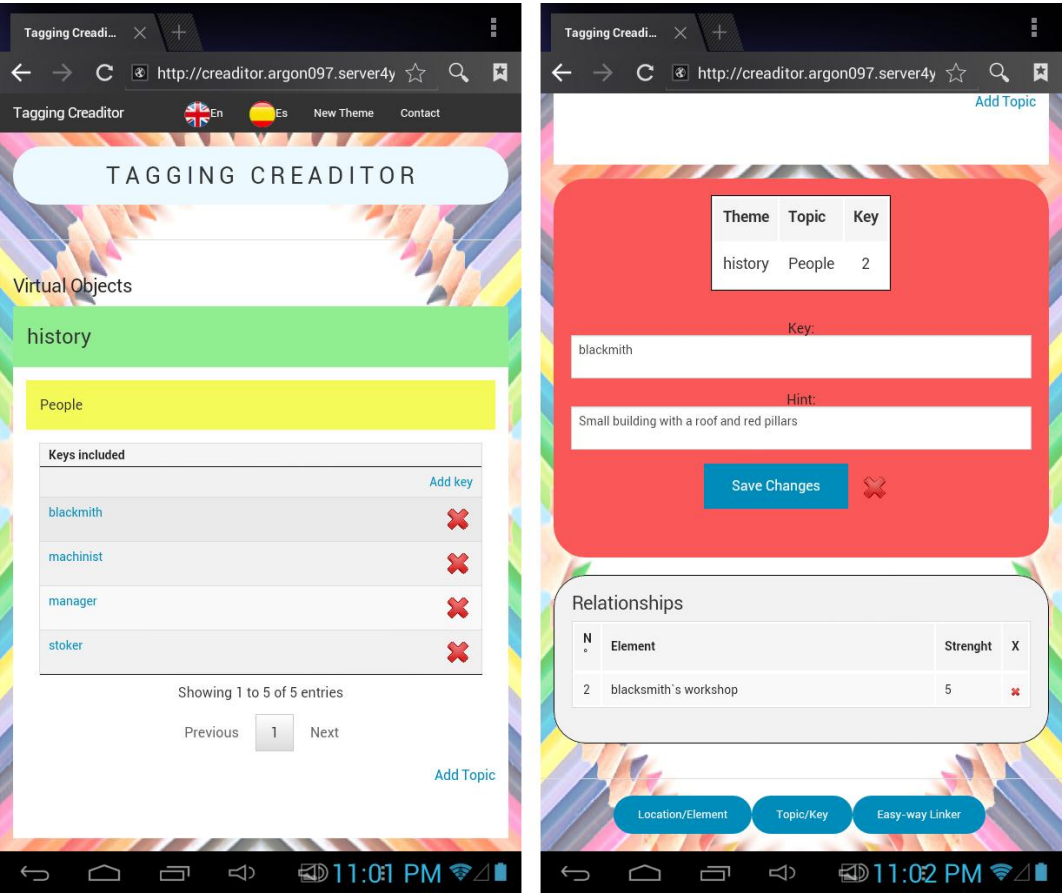


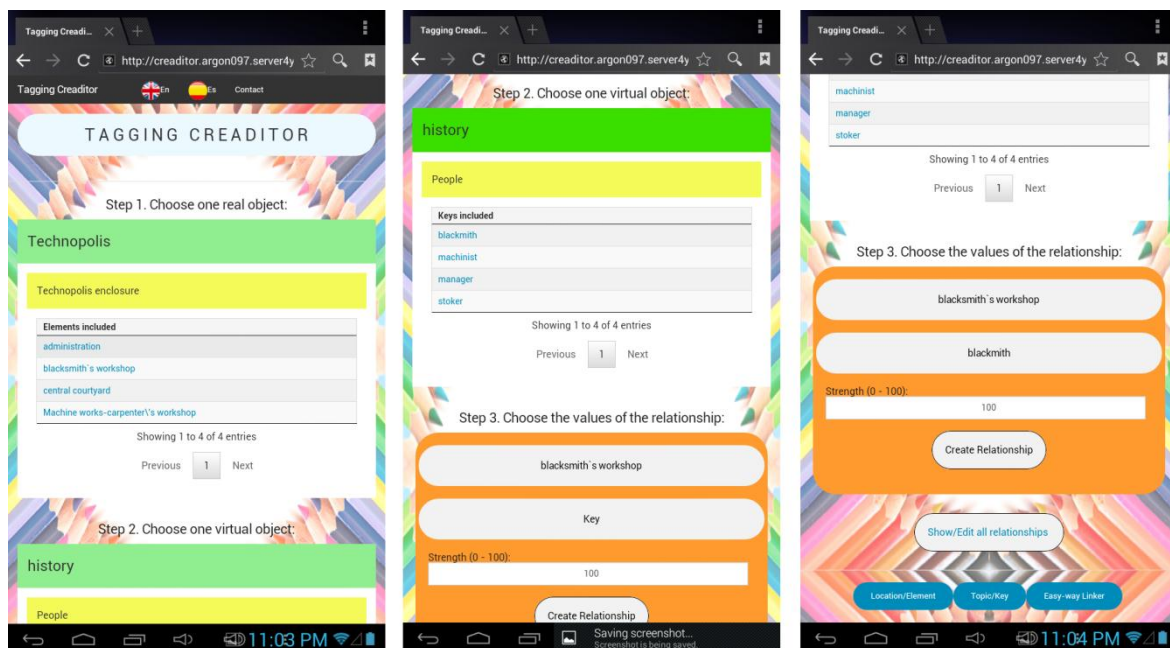
Figure 31. Virtual world management screen (left: navigation list, right: entity details boxes)

### Easy-way linker

As has been explained throughout the paper, the main purpose of mobile location-based games is to relate the existing physical entities of third level (i.e., elements) with existing virtual entities of third level (i.e., the keys) properly. Well, the main mission this screen achieves is to create relationships between the physical and virtual entities of third level.

The screen consists of three simple areas indicating three easy steps that the user has to follow:

- Navigation list of physical entities: This is the first step the user has to carry out in order to create a relationship and the purpose is to choose a real object in the navigation list provided.
- Navigation list of virtual entities: This is the second step the user has to carry out in order to create a relationship and the purpose is to choose a virtual object in the navigation list provided.
- Creator box of relationships: After choosing both physical and virtual objects, in this box the user will be able to check if the objects were chosen properly and to add the properties of the relationship such as the 'strength'.



**Figure 32. Easy-way linker screen in three simple steps: choose real object, choose virtual object and create relationship with the chosen values.**

### 4.3.2. Navigation through the main screens

In the Figure 33 we can see the diagram explaining how to navigate through the screens of the application.

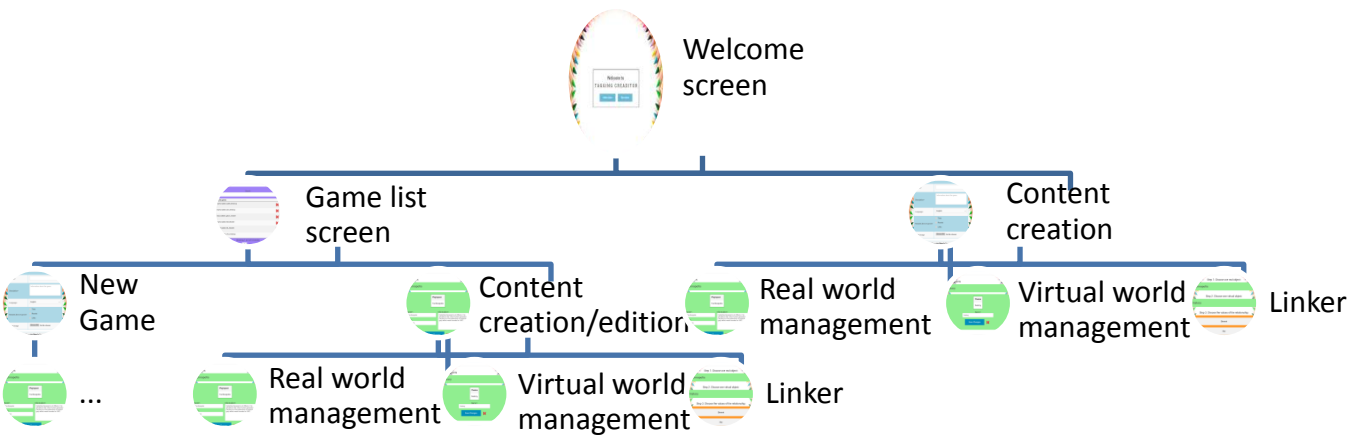


Figure 33. Navigation through the main screens of TaggingCreaditor.

## 4.4. Usage scenarios

In this section I will focus on describing some of the scenarios in which the application could be useful and also give an example used during the evaluation session.

First, note that TaggingCreaditor is not an application that people will use in general, since it is aimed at creating content for games in museums and locations of interest, at least, in touristic places.

For this reason, an interesting use could be that a school teacher prepares content for their students for a visit to a museum, a park or even a zoo. Another possible scenario is that museum workers prepare the contents of the museum providing visitors the option to play any of the location-based during the visit.

Another scenario could be that one of the developers of a location-based game wished to operate his game in a different location to where it is already created, or simply in the same location but with a different content adapted to a special date, as might be the Carnival of Patras.

In the next section we will see an example of a possible scenario with some schemes of content that should create the user.

### 4.4.1. Mobile example<sup>16</sup>

The description of the mobile scenario is as follows:

*“You are a history teacher in a school of Patras and your students and you are going to the Technopolis Museum in Athens next week. The best way for your students to learn knowledge about it is playing CityScrabble with your own content. Imagine that you are right now in the Technopolis Museum in Athens with the purpose of creating content for these games with a tablet. What you have to do, then, is to add content with the TaggingCreaditor editor at the same time as you go around the museum. The steps to do so are as follows:*

*1-Create a new game.*

*2-Enter real data. Put the data of the playspaces, locations and elements in the Location/Element screen.*

*3-Enter virtual data: Put the data of the themes, topics and keys in the Topic/Key screen.*

*4-Create the relationships between the elements and the keys in the Easy-way Linker screen.”*

---

<sup>16</sup> Scenario taken from the evaluation session documents.

After describing the scenario, the following step is to create the content. Firstly, the game must be created. To do it, the user will have to take into account that CityScrabble has two additional attributes that needs to work properly: radius for elements and hint for keys.

Secondly, the user has to create the real content and, to do it, the user will use the real management screen. As we are in a mobile scenario, when the teacher is adding elements to the editor, photos of the elements must be taken with the mobile device. One example scheme of the real content could be as it is represented in Figure 34.

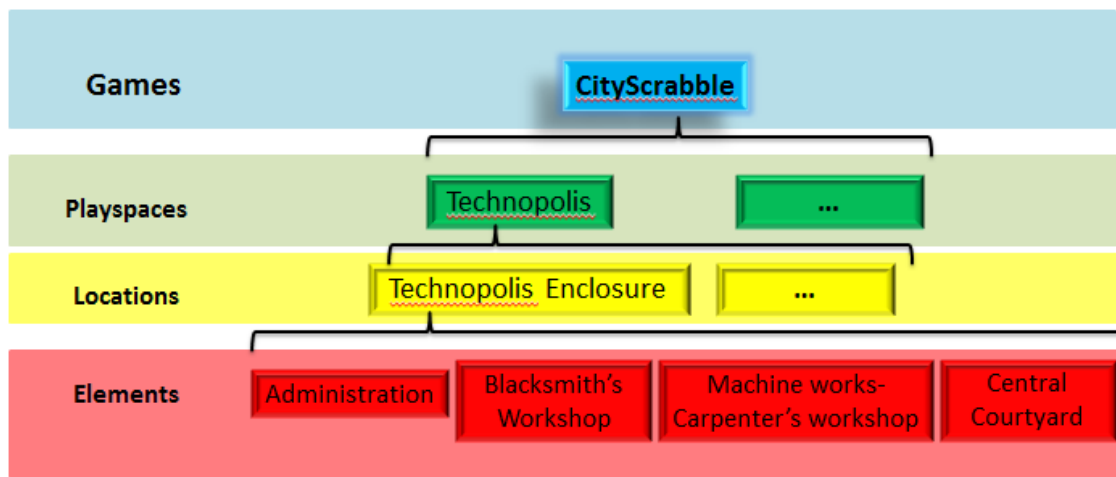


Figure 34. Example of real content that a teacher can create for CityScrabble.

The following step is to create the virtual content in the virtual management screen. On example scheme of the virtual content for the described scenarios could be the one that is represented in Figure 35.

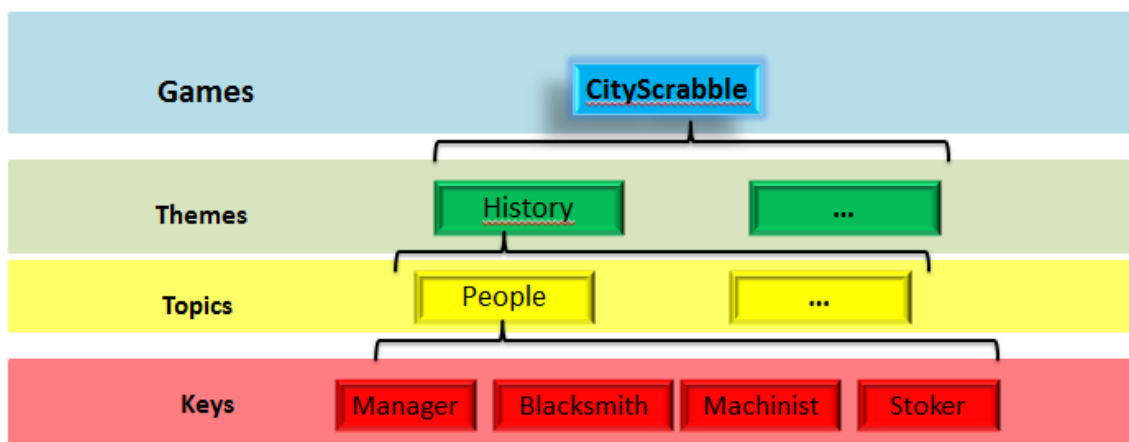


Figure 35. Example of virtual content that a teacher can create for CityScrabble.



The final step is to create the relationships between elements and keys in the linker screen. For the example I am proposing, the relationships should be:

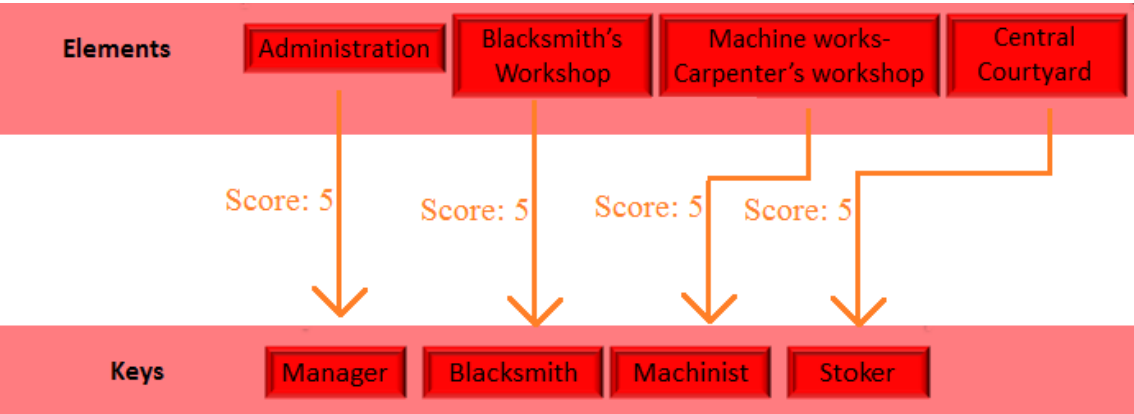


Figure 36. Example of relationships that a teacher can create for CityScrabble.



## 5. EVALUATION AND RESULTS

---

In this chapter I am about to present TaggingCreaditor assessment session and the results will be analyzed in detail.

Therefore, this chapter provides information on the preparation of the evaluation session, the necessary materials, the structure followed, the protocols and methods used in order to collect useful information from participants, problems found along the session along with their possible solutions and, finally, potential improvements that could be made to create much more efficient TaggingCreaditor application.

## 5.1. Introduction

The evaluation of the application took place on July 6, 2014 in the laboratory of the "Human-Computer Interaction Group" research group, University of Patras, Greece.

The evaluation involved 9 people of different nationalities who were responsible for evaluating the TaggingCreaditor tool. In addition to the participants, researchers that we have collaborated in the development of the application along with four researchers who helped under our guidance were in the assessment session. Participants created content in both the mobile version and the desktop version for two games that have been used to create the design of the application TaggingCreaditor: CityScrabble and Taggling. I will focus on explaining the assessment and the results of the mobile version.

The main tools that were used during the evaluation session were the Think-Aloud Protocol, a SUS (System Usability Scale) questionnaire and a moderated discussion between the testers and the organizers at the end of the session. All these tools and methods, which will be treated in more detail later, are especially useful for collecting information.



Figure 37. Photo taken during the assessment session.

## 5.2. Before the evaluation

Before explaining how the TaggingCreaditor assessment was developed it should be detailed how the session was prepared. Therefore, in this section we will see the preparations for the session: methods of collecting information, material and documents necessary and so on.

The point we should start to make a good assessment is to have an appropriate number of people. In this case, we got contact with nine people from different countries, the majority of them Erasmus in Patras, and all of them university students.

The next step is to know what the assessment session consisted of, as well, creating a scenario in which participants can work. In our case, the involved people were asked to complete several tasks for which they had to imagine they were school teachers in Patras. For the mobile version, the task was to create content simulating the tester was in a museum and had to create content in the museum. Therefore, the preparation of a room where there were replicas of the real museum exhibits hanging on the wall was required. These replicas were pictures of the real exhibits along with the respective descriptions. Therefore, we also needed to prepare the replicas of the museums that the participants needed to create content. The chosen museums were: Technopolis Museum in Athens and Museum of Contemporary Art in Thessaloniki.

When carrying out an assessment, it is important to ensure that all participants have the same information before starting to create content with the application. To do this, we decided to create a document that will contain all the information that participants needed to know, together with the relevant initial explanations. I also decided to record a tutorial video in order to show it in the room. The purpose of this video was the participants understand how to use TaggingCreaditor mobile application.

For the assessment session, we should prepare a form in which the participants can put their personal data that may be of interest (age, sex, education ...). In addition, the people must be reported that they will be recorded and that the information about them will be completely anonymous and just will be part of my work. The form and important information that participants should know are included in a document that will be described in the next subsection.

To collect data and to analyze them later on, several methods were used. The first is through the recording with a video camera, the second through the voice recording of the participants with a program included in the tablets. Both methods allow us to obtain all impressions and reasoning of the participants during the assessment since they were asked to use the Think-Aloud Protocol, which consists of observing a user working with an interface while encouraging them to "think-aloud"; to say what they are thinking and wondering at each moment. As a third method of collecting information I can include taking notes while the participant behavior is observed. In

addition to this, information may be obtained through SUS (System Usability Scale) questionnaires that the testers had to fill after using TaggingCreaditor to create content.

Therefore, as it has been indicated, the Think-Aloud Protocol, the SUS questionnaire and a moderated discussion among all participants of the meeting, including participants and organizers, were used during the course of the evaluation.

### 5.2.1. Documents <sup>17</sup>

Taking into account all that I mentioned above in addition to the digital content needed, the documents which were prepared for the assessment session are the following ones:

- Initial information document: This document is titled '1.Evaluation of TaggingCreaditor' and it is the first document located in the annex 2. The document includes: a form for participants to fill their personal information; initial information indicating to the participants that the information for the analysis would be processed anonymously, that they were to be recorded and during the performance of the tasks they had to use the Think-Aloud Protocol; an introduction of what the TaggingCreaditor application is and what its main purpose is; information about how CityScrabble is played; the two scenarios in which the people worked (desktop and mobile); and, finally, a questionnaire for the desktop version and another one for mobile version.
- Documents with instructions the testers followed. For the mobile version, the documents with the instructions are titled '2.CityScrabble-Instructions for the mobile scenario' and '3.Tagging-Instructions for the mobile scenario', found in the annex 2.
- Documents with photos and descriptions of real objects of each museum.

---

<sup>17</sup> The main documents prepared for the evaluation session can be found in the annex 2 (Evaluation session documents). The replicas of the museum, which were also prepared, are not included in this annex.

### 5.2.2. Material

In addition to the above documents, it was also required the following material:

- Computer (or laptop) and projector to show useful initial information to participants.
- Two desktop stations for evaluating the desktop version.
- Two tablets for evaluating the mobile version. These tablets should have prepared a program to record the voice of the participants while creating content in the application.
- Camera and video camera to record the behavior of the testers during the evaluation session.

## 5.3. Think-Aloud Protocol

The Think-Aloud Protocol is a very perceptive and informative method and when used effectively can result in a lot of information, regarding an interface, the usability of a system and so on. Think-Aloud Protocol consists of observing users working with an interface while encouraging them to "think-aloud"; to say what they are thinking and wondering at each moment.

Think-Aloud Protocol is of particular value because they focus on the problems a user has; when the user is working without difficulty, direct observation (and hence the think-aloud protocol itself) is of very limited use. This is because the user is unable to communicate as fast as they think and act, unless a specific problem arises which slows them down. It is at these times when this method really shines as it allows the observer to correlate the actions and statements of the participant.

This protocol may be used in two different scenarios:

- The observer specifies a definite task to be accomplished by the subject. This allows the observer to concentrate on a specific task they are interested in.
- 'Open-ended'; no task is specified, and the user is free to choose their own task. Allows the observer to concentrate on naturally occurring problems.

In the evaluation session of TaggingCreaditor, we decided to choose the first scenario because we were interested in how the user creates content for location-based games.

### 5.3.1. Advantages of Think-Aloud Protocol

The main advantages of this method are:

- Rapid, high-quality, qualitative user feedback as compared with questionnaires.
- Data is available from a broad range of sources, such as: Direct observation of what the subject is doing and hearing what the subject wants, or is trying to do.
- If the subject gets into difficulties, the observer has the chance to clarify the situation.
- High degree of flexibility; the experiment may easily be steered by the observer.
- The presence of two people allows meaningful, direct dialogue.

## 5.4. SUS – System Usability Scale<sup>18</sup>

The System Usability Scale (SUS) is a simple, ten-item scale giving a global view of subjective assessments of usability. The System Usability Scale (SUS) was released into this world by John Brooke in 1986. It was originally created as a "quick and dirty" scale for administering after usability tests on systems like applications.

SUS is technology independent and has been made freely available for use in usability assessment and has been used for a variety of research projects and industrial evaluations, for instance, it has been tested on hardware, consumer software, websites, cell-phones, and so on. It has been proved to be a valuable evaluation tool, being robust and reliable. It correlates well with the subjective measures of usability.

### 5.4.1. Operation

The System Usability Scale is generally used after the respondent has had an opportunity to use the system being evaluated, but before any discussion takes place.

SUS is a 10 item questionnaire with 5 response options. It can be seen that the selected statements actually cover a variety of aspects of system usability, such as the need for support, training, and complexity, and thus have a high level of face validity for measuring usability of a system. The 10 selected statements are shown in the following box:

---

<sup>18</sup> References: [23], [24] and [25].

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The SUS uses the following response format:

<b>Strongly disagree</b>					<b>Strongly agree</b>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1	2	3	4	5	

**Figure 38. System Usability Scale response format.**

An important rule to be met is that all items should be checked. If respondents feel that they cannot respond to a particular item, they should mark the center point of the scale.

#### **5.4.2. Scoring and interpreting SUS**

The SUS scale is a Likert-style scale and its final result is a single value representing a composite measure of the overall usability of the system under study. That is, that at the end of processing the data, we have a unique value that can be in

the range between 0 and 100. This is what allows us to easily compare systems evaluated by SUS scale.

It should be noted that independent ratings are not significant in themselves. That is, if in the first question we had obtained a massive response from 'Strongly agree', is not a significant finding as it is isolated from the rest.

In order to score SUS, we have to follow these rules:

- For odd-numbered items: subtract one from the user response.
- For even-numbered items: subtract the user responses from 5
- This scales all values from 0 to 4 (with four being the most positive response).
- Add up the converted responses for each user and multiply that total by 2.5. This converts the range of possible values from 0 to 100 instead of from 0 to 40.

## 5.5. Focus group

Korman<sup>19</sup> defines the focus group as "a meeting of a group of individuals selected by researchers to discuss and elaborate, from personal experience, a thematic or social fact that is under investigation".

It is a method for collecting qualitative information that arose in the early 30s when scientists become aware of the problems presented by a traditional interview in which the interviewer could influence the answer of the interviewed person.

Recently, the group meetings are being widely used in evaluations of software (and in many other fields) because they provide wide information, at low cost and in little time.

The secret of the focus group lies in participants can freely express their views and opinions on different aspects.

The main characteristics of the focus group are:

- Participants present homogeneous characteristics.
- The data presented are qualitative.
- The discussion is focused on a specific aspect, in my case in the usability of TaggingCreaditor.
- The purpose is not to find consensus among the participants, but to establish perceptions, feelings, opinions and thoughts of individuals.

---

<sup>19</sup> Korman, Hyman. The Focus Group Sensign. Dept. Of Sociology, SUNY at Stony Brook. New York. 1986.



## 5.6. Evaluation session

The evaluation session was organized by the two developers TaggingCreaditor application (desktop version and mobile version) with the help of four researchers who followed our guidance.

The participants had to make teams of two or three, so that there were three teams of two and three-person team. During the evaluation session, two of the teams had to work with the desktop version, while the other two worked with the mobile version.

As a developer of the mobile version, I was responsible for observing and offer help to the teams that used the tablets to work in the mobile scenario described above. Likewise, the developer of the desktop version was responsible for providing support to the teams that used the desktop version.

With regard to the four supporting researchers, each one had a role. One of them was responsible for taking notes about the behavior and the problems the work teams were having; one of them was in charge of taking photos and videos of the session; another of them collaborated explaining the operation of Taggling; and finally, the last researcher moderated the discussion at the end of the evaluation session.

The evaluation session was structured into several phases:

1. Initial information phase.
2. Description of location-based games.
3. Scenario description.
4. Demonstration of TaggingCreaditor use.
5. Action phase.
6. SUS Questionnaire.
7. Focus group discussion.

Each of them will be explained in detail in the following sections.

### 5.6.1. Initial information phase

At the beginning, the testers of the session were informed about how the evaluation was going to be organized and an introduction on the TaggingCreaditor application was made. In addition to this, a document was given to each participant in which all the information needed was shown: Initial information document ('1.Evaluation of TaggingCreaditor' in annex 2).

### 5.6.2. Description of location-based games

After participants understood that the application can create or edit content to location-based games, we moved on to explain the operation of the game they had to create content. The first game for which they had to create content was CityScrabble and the second was Taggling.

### 5.6.3. Scenario description

The next step before the testers started to use TaggingCreaditor was the explanation of the scenario in which they had to work with the mobile application.

For CityScrabble, participants imagined to be history teachers from a school of Patras and had the objective of creating content for Technopolis Museum in Athens so that their students could play CityScrabble with that content in a future tour.

The process was very similar for Taggling except that participants had to create content for the Contemporary Art Museum in Thessaloniki, so that his students, again, could play Taggling with that content in a future tour.

In the mobile scenario both games, participants had to imagine that they were in the museum itself so they had to create content with the tablet taking pictures of the items placed on the walls of the room prepared for this purpose.

### 5.6.4. Demonstration of TaggingCreaditor use

After explaining the scenario, a demonstration through a tutorial video was showed in the projector. In this video was explained how to work and use TaggingCreaditor in a tablet.

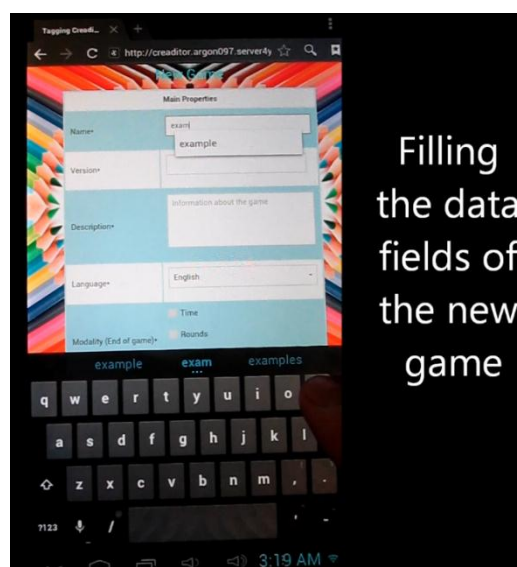
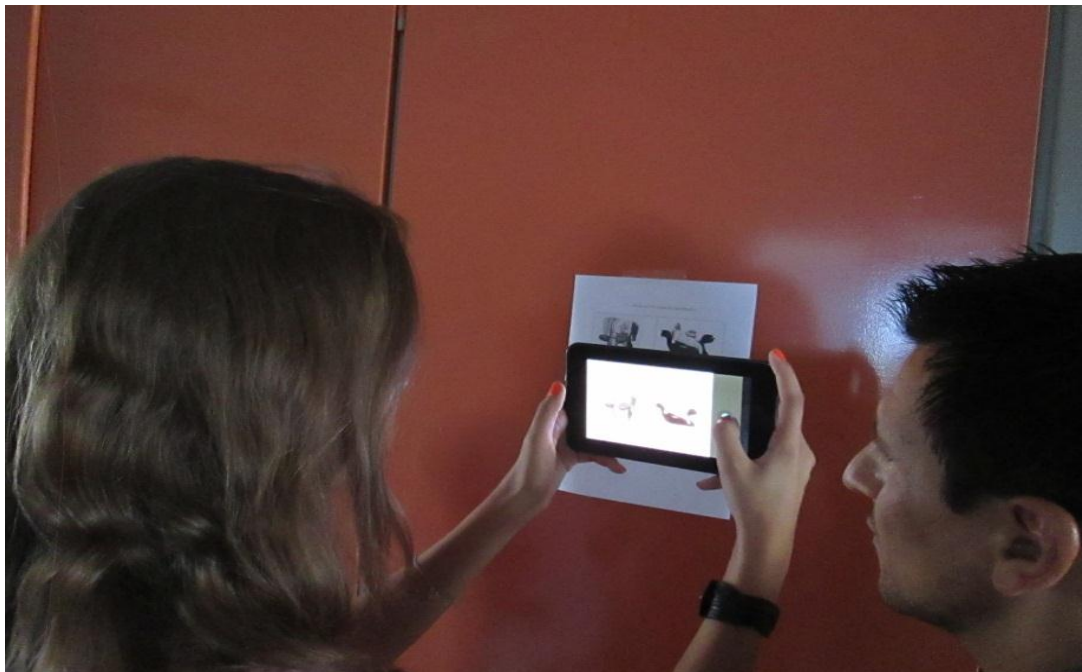


Figure 39. Screenshot of the video tutorial showed during the evaluation.

### 5.6.5. Action phase

The next step was to get to work creating new content, firstly, to CityScrabble, and secondly, to Taggling. To do this, they had to follow the steps outlined in the documents '2.CityScrabble-Instructions for the mobile scenario' and '3.Taggling-Instructions for the mobile scenario', found in Annex 2.



**Figure 40. A work team taking a photo of a replica while they are creating content.**

### 5.6.6. SUS Questionnaire

Once participants had finished creating the content that they were asked, the next step was to answer a SUS (System Usability Scale) questionnaire about the application.

We chose this type of questionnaire because it is an established questionnaire that can help to measure the usability of a system. In addition to this and because of this reason, using SUS has the advantage that we are able to compare our results with a lot of results of other evaluations.

### 5.6.7. Focus group discussion

Finally, an interesting discussion was held among all participants and organizers of the evaluation in which all participants commented their own impressions, opinions, problems they found, possible improvements and so on.



Figure 41. Focus Group discussion which took place in the end of the evaluation.

## 5.7. Results and conclusions

### 5.7.1. Results during the evaluation

Thanks to the voice recording of the participants and the use of the Think-Aloud Protocol, we can analyze their thoughts in order to draw conclusions about how the problems that the user was having could be solved.

By observing and taking notes of how the user behaved while creating content on the scenarios of the testing session, I could see some of the bugs or errors in the user interface that the application has.

Furthermore, the SUS questionnaires will be really useful to analyze the usability of the system and, on the other hand, the focus group will help a lot taking the opinions of all the participants and evaluators. These last two methods of getting information during the evaluation session will be detailed in the next two sections.

During the evaluation session was observed that the user expected behaviors of the application that did not happen. For example, when creating a key, the message appeared that the key was successfully created, but then the key was not on the list of

navigation of virtual objects until the page of the application was not refreshed. This problem led the user to create the same key twice for example.

Another problem that most participants found was that they were lost when they had to search the buttons to add a new location, element, topic or key. This happened because no one told them that in order to create a location, for example, they had to click on the playspace in which they wanted to add the location. Following problems with the user interface, the first time they had to switch screens all had problems in the mobile version, they did not know how to do it because the navigation buttons are located in the lower part of the screen. These problems can be solved with the mobile user manual included in Annex 1.

### 5.7.2. Analysis of SUS questionnaires

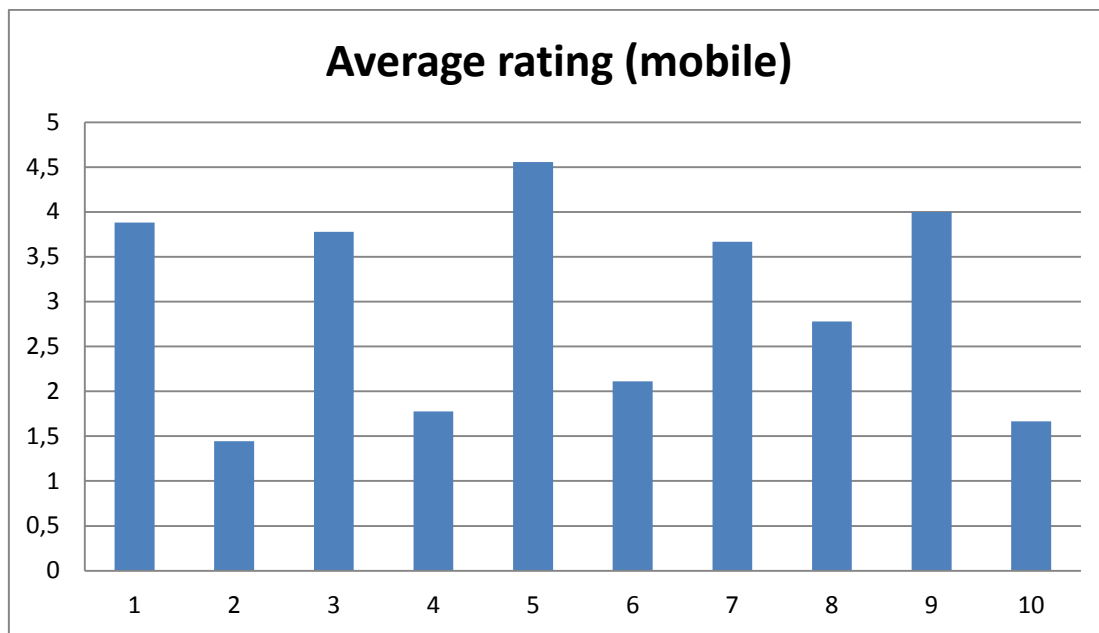
Firstly, I will focus on my part of the work, the study of the assessment to the mobile version of the application.

To make a synthesized analysis of the questionnaires, it is best to work with the average score of participants surveyed in each of the statements to study for the mobile application.

In the following table we can see the resulting average rating of the participants for the mobile version. In the left column are represented the ten statements that formed the questionnaire, and the right column the mean scores of each statement on a Likert scale of 1 to 5 defined.

STATEMENT	AVERAGE
1	3,88
2	1,444444444
3	3,777777778
4	1,777777778
5	4,555555556
6	2,111111111
7	3,666666667
8	2,777777778
9	4
10	1,666666667

More visually see these scores in a bar graph:



**Figure 42.** Graph showing the average rating of the SUS questionnaire for the mobile version.

To comment the graph in a general way, I will analyze the ratings that are more interesting. As we can see, people have given top marks to the fifth statement, so it could be said that, in general, participants have found that all the functionalities of the application were properly developed. Followed by this, it is noteworthy that the second highest score corresponds to ninth statement, i.e., individuals felt very confident using the system.

The aspect that has had the lowest score was the second statement; individuals did not find the system unnecessarily complex, which turns it into something positive, the system showed no unnecessary complexity.

Moreover, respondents said they do not need to learn many things before using the system, which could be translated as something positive because more people will be able to access the mobile editor.

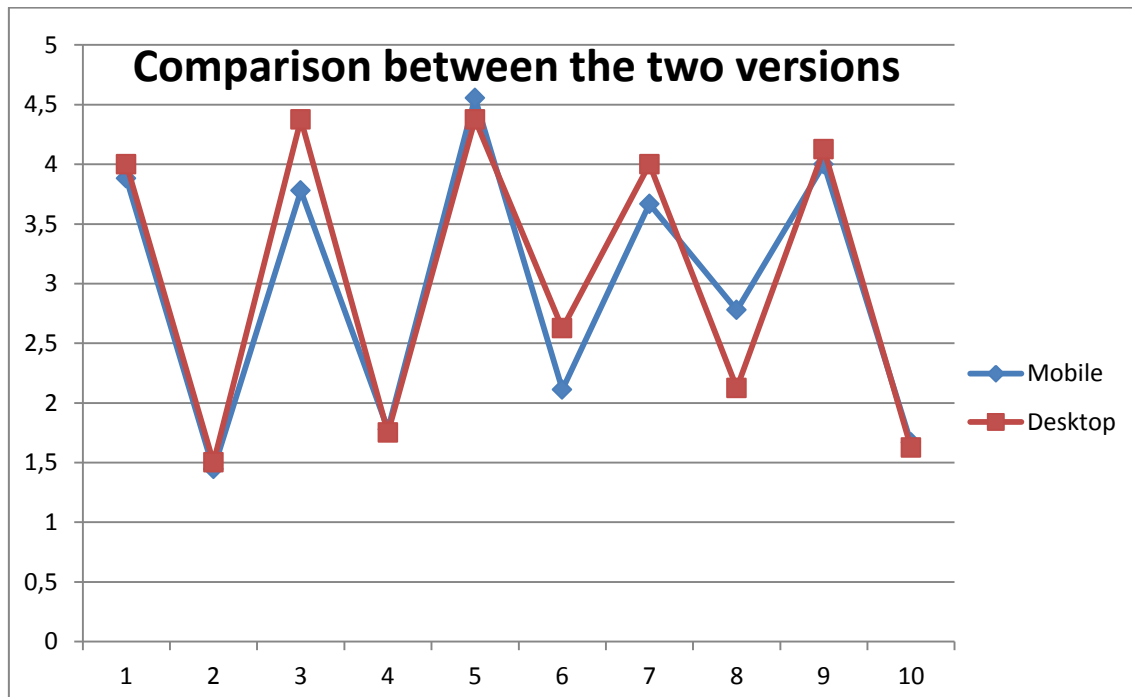
I believe that one possible way of enriching the study is to compare the scores corresponding to the desktop version and the mobile version and, thus, make a small approach on what differences the users found between the two versions.

The following table shows the average scores of individuals to the statements in both the mobile and the desktop version.

It should be mentioned that one of the participants did not answer the survey corresponding to desktop, so the mean scores of this version is divided by eight while in the mobile version is divided by nine.

STATEMENT	MOBILE AVERAGE	DESKTOP AVERAGE
1	3,88	4
2	1,444444444	1,5
3	3,777777778	4,375
4	1,777777778	1,75
5	4,555555556	4,375
6	2,111111111	2,625
7	3,666666667	4
8	2,777777778	2,125
9	4	4,125
10	1,666666667	1,625

To compare the rating of both versions, it is much easier with the chart below.



**Figure 43. Chart showing a comparison between the average ratings of the mobile and desktop versions.**

First of all, it must be highlighted that, practically, there are no significant differences between the two versions. However, I will comment the most remarkable data.

The main difference is in the eighth and the third statements. In general, the respondents believe that the mobile version is more cumbersome and difficult to use than the desktop.

About the different average score of the sixth statement, I can conclude that testers believed that the desktop version is more inconsistent than the mobile version.



Moreover, it is worth to mention that according to the mean scores, the desktop version is easier to learn than the mobile. However, this rating is due to the sample is biased because if we look at the individual ratings in each of the versions, we would get the conclusion that the mobile version is thought to be more easily learned by users.

Regarding the fifth statement, testers usually say that the functionalities in the mobile version are better integrated than in the desktop version.

To conclude with this section, it is really interesting to know which the SUS scores are for both the mobile and the desktop version. The way to calculate it is following the rules that were established in the SUS section: for odd-numbered items, subtract one from the response; for even-numbered items, subtract the user responses from 5; and, finally, add up the converted responses for each user and multiply that total by 2.5. I have used the average scores calculated during the study of the results. The results are the following:

SUS SCORE (MOBILE VERSION)	SUS SCORE (DESKTOP VERSION)
75.28	78.128

This confirms what I have said before that, practically, there are no significant differences between the two versions.

### 5.7.3. Analysis of the focus group

The general opinion of the people was that the application was of interesting use and they valued very positively that the same application, TaggingCreaditor, could create content for multiple games, in their case for MuseumScrabble and Taggling.

During the focus group meeting some small bugs and interface problems were discussed. With a future update they can be easily solved.

When participants were asked about the differences between the mobile version and desktop version, all of them found much more interesting, useful and fun the mobile version because when creating content they have to interact with the mobile device, for instance, taking pictures of the real exhibits. In addition, for most of them the user interface of the mobile version was more intuitive, although the desktop version was easier to create content because the typing process was faster.

One point mentioned by several of the testers was that at first they were a bit lost in the user interface, i.e., they would have liked to have more guidance or explanations, especially in the early times when the application was used. The problem



of being lost at the beginning of the use could be solved by providing the user manual found in annex 1. On this point, they also indicated that after accustomed to the use of the application, the process of creating content was much faster and easier.

To conclude, some possible improvements that a possible future version would include were discussed, such as the mobile version got the current location of the mobile device automatically, speech recognition, or even develop a simpler interface.

---

## 6. CONCLUSIONS

---

The idea of developing this application arose from an existing tool that allowed the creation of content for the location-based game called CityScrabble, created by Ruben César Llorente. From this idea, we realized that there was no tool able to create content in general, that is, that enables the creation of content for several location-based games that share a lot of features in common.

After the presented work in this paper, it can be said that the mobile application, aimed at creating or editing content for collaborative location-based mobile games, is working properly and, therefore, it has managed to meet all the initial requirements and key objectives. After the development of the application, I have been aware of potential improvements and new functionalities that could be added in a more advanced version of the content editor.

To begin with, among the improvements that could be carried out in the editor in order to achieve a much more complex final editor I can mention, for example, the adaptation of the application to multi-user support or the support of the greek language in the user interface.

Secondly, very interesting functionalities could have been added in order for the content editor to be far more flexible. One of them is the option to introduce a drawing of a museum map so that the content creator could locate the physical elements he wants on the map or, as a variation, that the map of Google Maps used has greater accuracy when positioning objects in indoors buildings. A second interesting feature, that would make many more location-based games could use this tool, is that the user could select the object hierarchy he wants, that is, other than the fixed hierarchy to which the user has to meet. As a final example of new functionality, it would also be very useful if the user could upload files (for instance, XML files) in which the content of a game is found, so that when the user uploads it, all data is entered in the editor database automatically.

To conclude, after having met all the requirements established initially and the main objectives of the project, I can say that there is now a mobile application able to create content for a variety of collaborative location-based mobile games and that has the features of being a very flexible, generic and collaborative tool with a powerful potential to be enhanced in a future version by adding new functionalities.

## 7. REFERENCES

- [1] Nikoleta Yiannoutsou and Nikolaos Avouris, “Reflections on use of location-based games playful narratives for learning”, Human Computer Interaction Group, University of Patras, Greece.
- [2] Nikolaos Avouris and Nikoleta Yiannoutsou , “A review of mobile location-based games for learning across physical and virtual spaces.”, Human Computer Interaction Group, University of Patras, Greece.
- [3] Christos Sintoris, Adrian Stoica, Ioanna Papadimitriou, Nikoleta Yiannoutsou, Vassilis Komis, Nikolaos Avouris, “MuseumScrabble: Design of a mobile game for children's interaction with a digitally augmented cultural space.” Department of ECE, HCI Group, Department of ESECE, ICTE Group, University of Patras, Rio Patras, Greece.
- [4] Marco Antonio Munoz Alonso, “Design and development of a mobile city game: CityScrabble”, Human Computer Interaction Group, University of Patras, Greece.
- [5] <http://hci.ece.upatras.gr/museumscrabble/>, Last access: June 2014
- [6] <http://www.cityscrabble.gr/>, Last access: June 2014
- [7] <http://hci.ece.upatras.gr/bms/index.php?id=screenshots&lang=en>, Last access: June 2014
- [8] Jean E. Sammet, “Programming Languages: History and Future”, IBM Corporation.
- [9] [http://www.inf.unideb.hu/kmitt/konvkmitt/programming\\_languages/book.xml.pdf](http://www.inf.unideb.hu/kmitt/konvkmitt/programming_languages/book.xml.pdf), Last access: June 2014
- [10] Ánton Rodríguez, M. & Pérez Juárez, M. Á., “Tecnologías para Aplicaciones Web 2013-2014, Programación en PHP”, E.T.S.I. de Telecomunicación, Universidad de Valladolid, December 2013.
- [11] Grupo EIDOS, “Lenguaje HTML”, Versión 1.0.0, 2000 ([www.lalibreriadigital.com](http://www.lalibreriadigital.com))
- [12] Ánton Rodríguez, M. & Pérez Juárez, M. Á., “Tecnologías para Aplicaciones Web 2013-2014, JavaScript”, E.T.S.I. de Telecomunicación, Universidad de Valladolid, October 2013.
- [13] <http://jquery.com/>. Last access: June 2014
- [14] Ánton Rodríguez, M. & Pérez Juárez, M. Á., “Tecnologías para Aplicaciones Web 2013-2014, Cascading Style Sheets”, E.T.S.I. de Telecomunicación, Universidad de Valladolid, October 2013.
- [15] Ánton Rodríguez, M. & Pérez Juárez, M. Á., “Tecnologías para Aplicaciones Web 2013-2014, Ajax”, E.T.S.I. de Telecomunicación, Universidad de Valladolid, October 2013.

- [16] <http://foundation.zurb.com/>. Last access: June 2014
- [17] <http://zurb.com>. Last access: June 2014
- [18] <http://www.mysql.com/about/>
- [19] <http://www.mysql.com/products/workbench/>
- [20] <https://httpd.apache.org/>, Last access: June 2014.
- [21] [www.apache.org](http://www.apache.org)., Last access: June 2014.
- [22] [www.phpmyadmin.net](http://www.phpmyadmin.net), Last access: June 2014.
- [23] J Brooke, “SUS-A quick and dirty usability scale”, Usability evaluation in industry, 1996.
- [24] A Bangor, PT Kortum, JT Miller, “An empirical evaluation of the system usability scale”, Intl. Journal of Human–Computer, 2008.
- [25] JR Lewis, J Sauro, “The factor structure of the system usability scale”, Human Centered Design, 2009.

# ANNEX 1.MOBILE USER MANUAL

In this section, I will explain in detail how to work with the main functionalities of the mobile editor TaggingCreaditor. During my explanations, I will employ the intuitive terminology of the entities hierarchy, i.e., for the real world I will use the terms playspace, location and element; while for the virtual world I will use the terms theme, topic and key.

The common scenario follows the following steps:

1. Creation of a New Game.
2. Creation of the real objects of the game.
3. Creation of the virtual objects of the game.
4. Creation of the relationships between the real objects and the virtual objects.

Each one of these steps is carried out by the user in different screens, as we will see during this manual.

## I. Welcome screen

This screen simply welcomes the user of the application and gives the options 'Select Game' or 'Create Game'. In the first option the user can check or edit the content data of a previously created game and the second option will allow the user to create content for a new game. We will see, firstly, the common scenario: creating a new game.



Figure 1. Welcome screen.

## II. New game

When creating a new game we will find three different areas or blocks, as can be seen in Figure 2:

-Main properties area: In this area the user can enter the most general characteristics of the game. Among them are: name, version, description, language, modality and game image.

-Hierarchy object's names area: In this area the user can change the default names of the entities in both the real world and the virtual world for creating and editing content. The default values for the real world are Playspace (first level), Location (second level) and Element (third level); while for the virtual world: Theme (first level), Topic (second level) and Key (third level). These fields are really useful and give to the editor a great flexibility since the content editor user will be able to choose the names of the entities in the hierarchy so that, for example, if someone wants to create content for CityScrabble, the real object of third level would be 'objects', whereas, if someone wants to create content for MuseumScrabble, the real object of the third level would be 'exhibits'.

-Plugging properties block area: Some types of games need some extra properties in their content in order to work properly. In this area the user can select the additional properties that the user wishes in order to add them to the later creation of content. Among the optional properties that can be added: radius for the element entity; difficulty for the topic entity; and, finally, visibility, hint and type for the key entity. All of them have been explained in the design of the database.

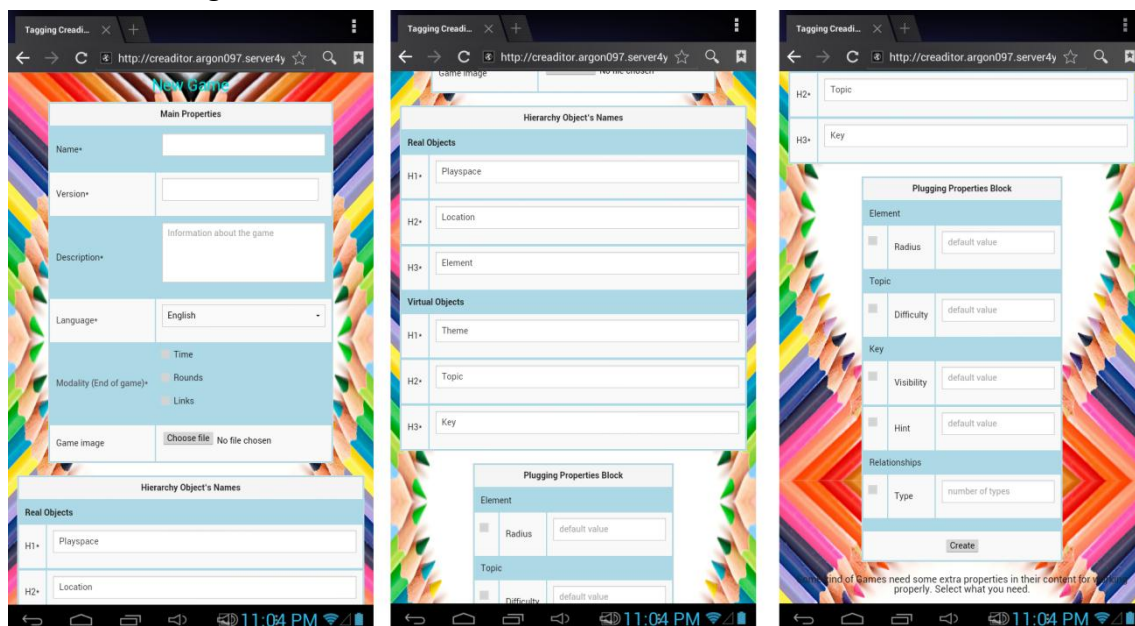


Figure 2. New game consists of three blocks: main properties, hierarchy objects' names and plugin properties block.

### III. Management of the real world entities

The following step after creating a game is to create the real objects that will be part of the game. The user will be able to do this in the real world entities management screen.

When managing the real-world entities, we have three options for each: create entity, edit entity or delete entity. To manage real world entities, we will have to do on the screen I called real world management screen above. We will be seeing each of the cases of creation, and then I will explain the case of editing and deleting with an example, because it is applicable to all entities.

In the following figure, you can see where the creation buttons for the real objects are located in the user interface:

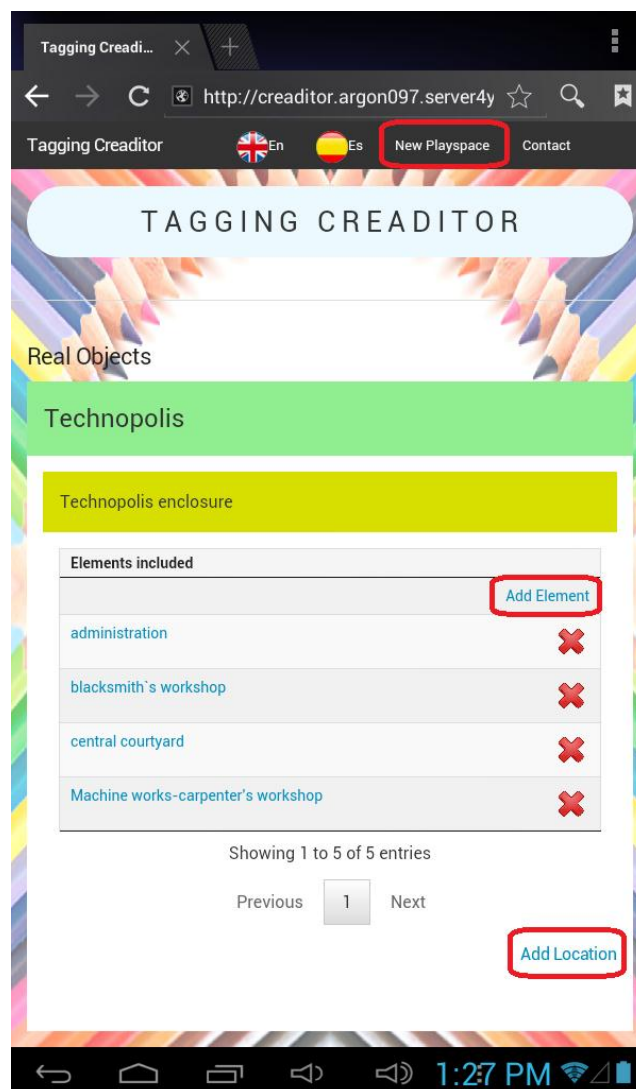


Figure 3. Real world management screen and its creation buttons.



### Creating a new playspace

To add a playspace we have to go to the top bar menu of the real world entities management screen. After selecting this option we will see the creation screen of playspace. In it, as you can see in Figure 4, the fields name, kind and information must be filled to create a new playspace or, as second option, it is offered to the user to add an existing playspace. If the user chooses to add an existing playspace, the user will see the second screen of Figure 4, being able to choose either add an existing one or clone an existing one. The difference is that adding an existing one, by modifying features or adding objects to this playspace, this playspace in the other games will be modified (i.e., the same playspace may be modified in whatever the game containing this playspace); while cloning an existing one means that you copy all the features and objects found in the playspace and are copied to the game, that is, if the user changes something in the cloned playspace, it will not be changed in the origin playspace.

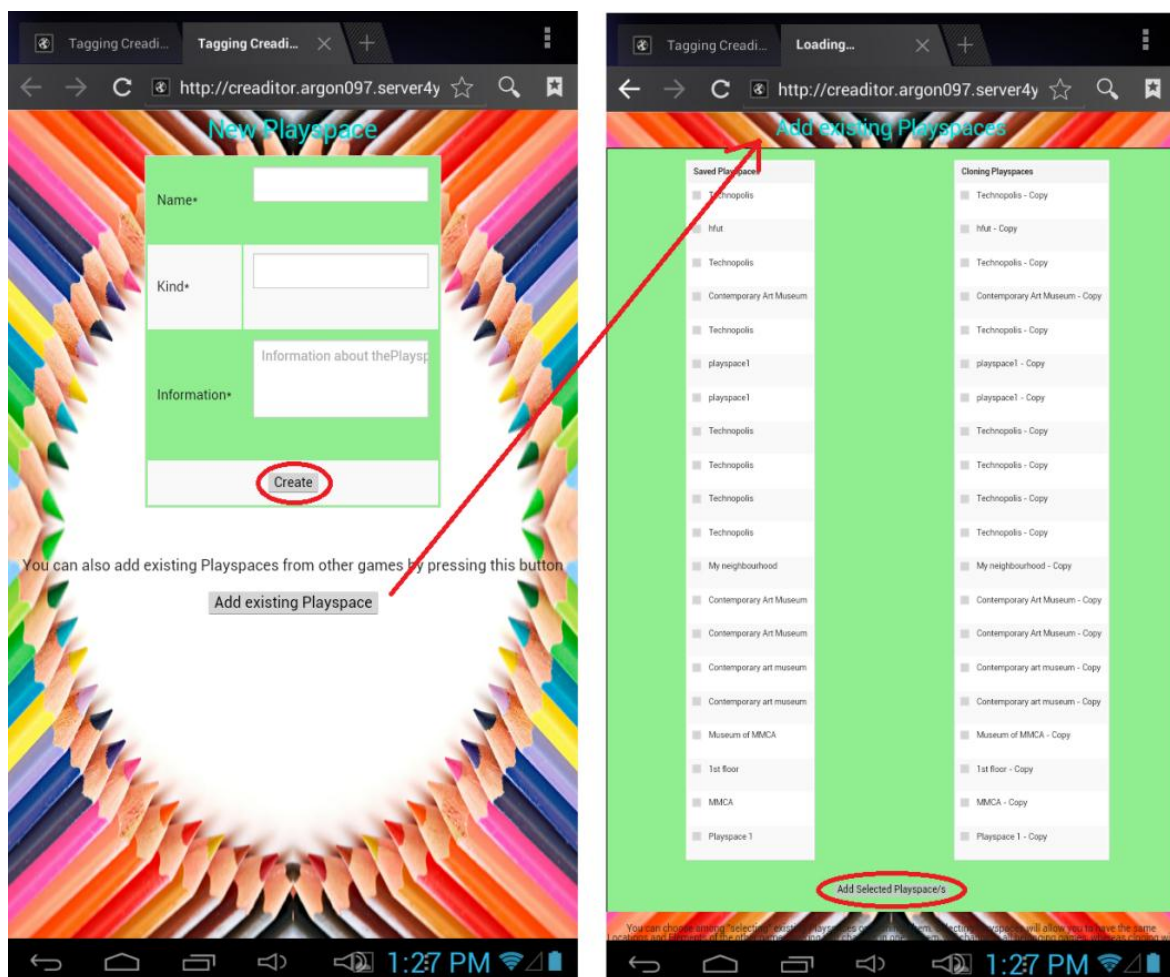


Figure 4. Creating a new playspace or adding an existing one.



### *Creating a new location*

To add a new location to an existing playspace, the user firstly needs to choose in the list of objects the playspace to add the location the user wants to create. By choosing it, the locations that are part of playspace, if there is any, and the option of adding a new one will appear. Once the user chooses the option to add location, creation screen of location appears. In it, the user will fill the fields name, type and information and the location will be created.

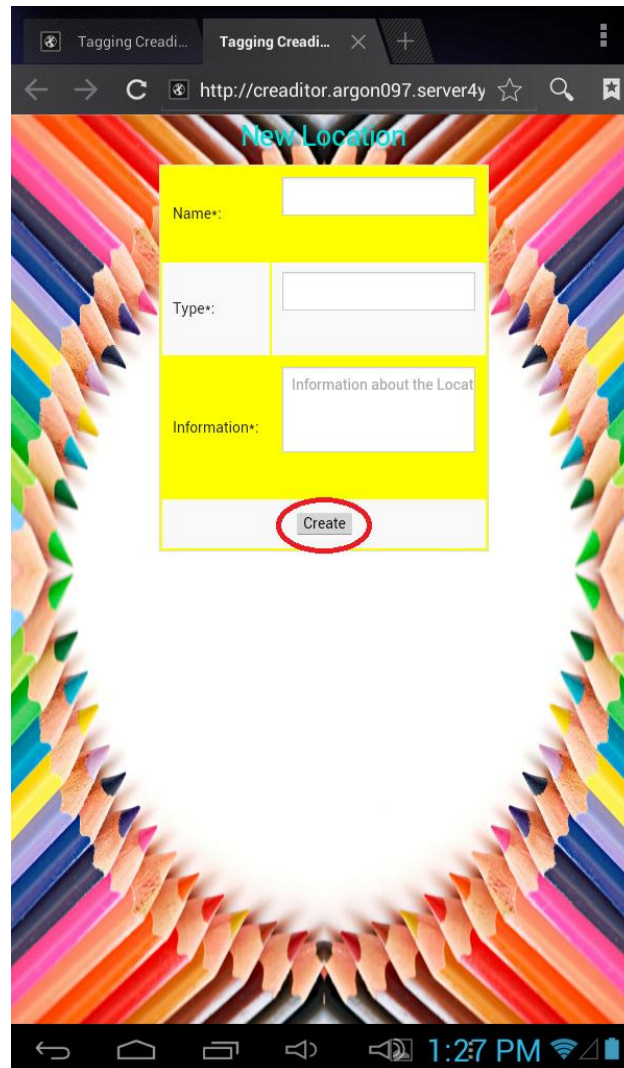


Figure 5. Creating a new location.

### Creating a new element

To add a new element to an existing location, the user firstly needs to choose from the list of objects the location to add the element the user wants to create. By choosing it, a list of elements that are part of the location, if there is any, and the option to add a new one appears. Once the user chooses the option to add element, the creation screen of element appears. In it, the user should fill in the name field, choose a picture to identify the element, fill in the information field, place the item on the map and add a radio in case the user has decided to add this optional property. After that, the new element will have been created.

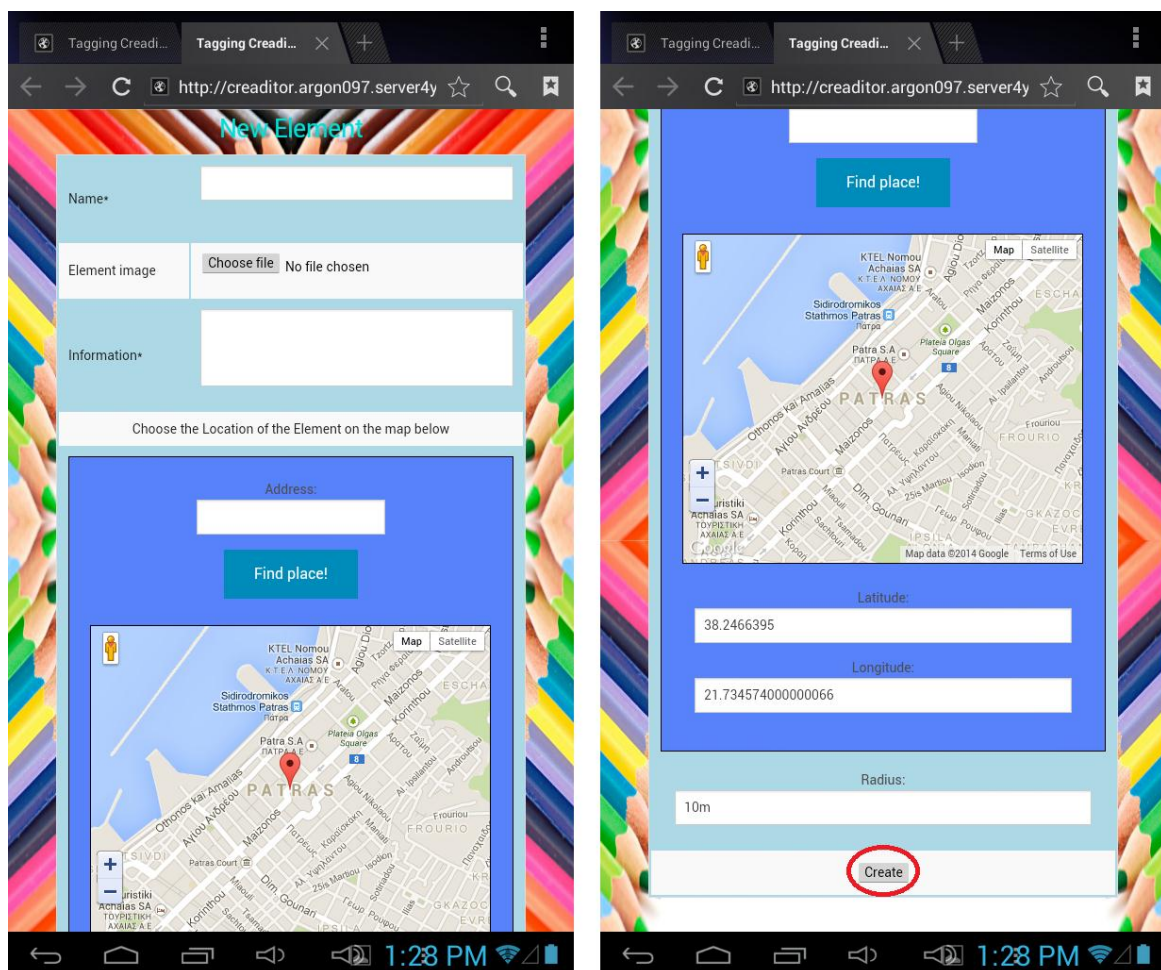


Figure 6. Creating a new element.

### *Editing and deleting the real world entities*

To edit any of the major attributes of a real world entity what first the user has to do is to select the entity to modify from the list of objects. The details of this entity appear, then, in the detail area, and what the user needs to do is simply modify the desired values in the area in which the details are displayed and press the 'Save Changes' button.

To delete any of the real world entities what first the user has to do is to select the entity to delete from the list of objects. After this, the user simply presses the image with the red cross of the area in which the details are shown and confirm you wish to delete the entity. A more quickly way when removing elements is provided in the application: the option is given to the user to delete elements from the list of objects, so that the user doesn't have to select the element before.

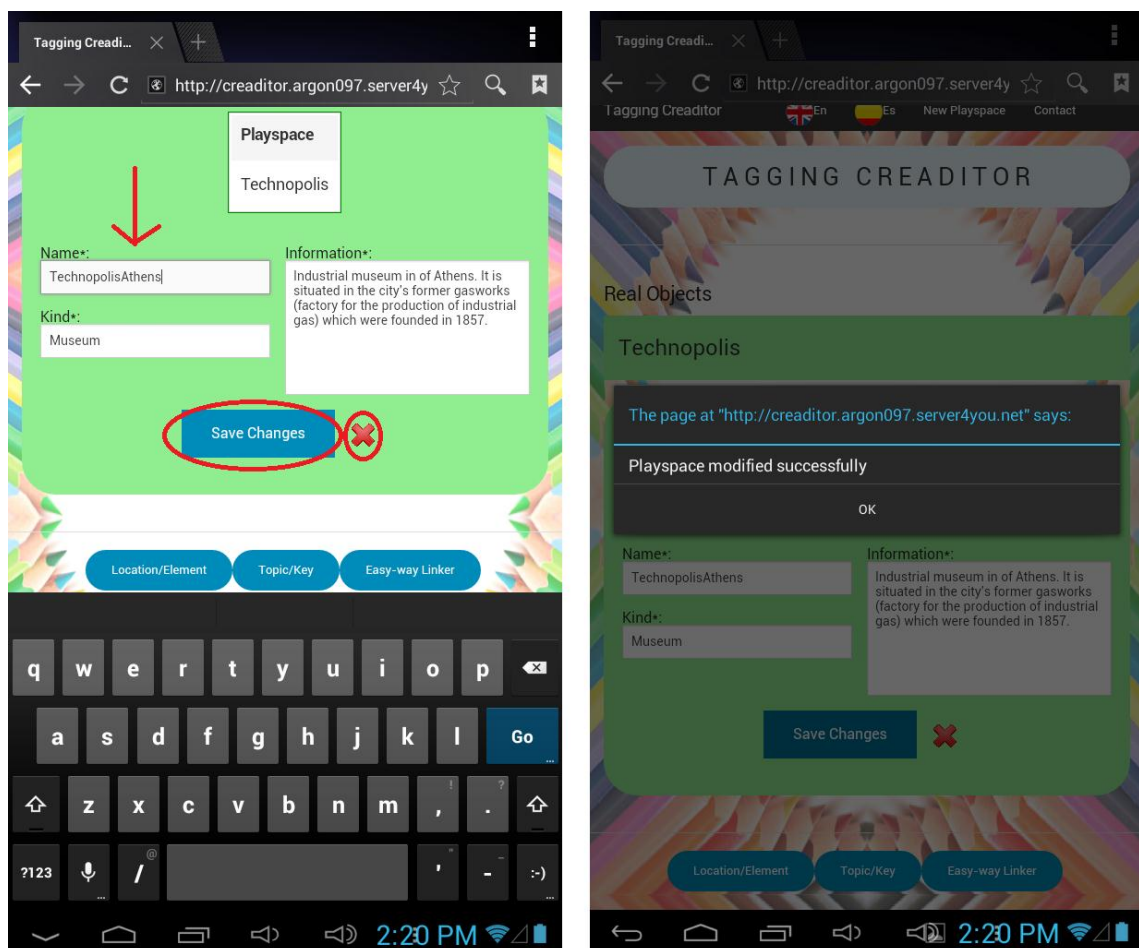


Figure 7. Editing a playspace to show an example.

#### IV. Management of the virtual world entities

The following step after creating the real objects of the game is to create the virtual objects that will be part of the game. The user will be able to do this in the virtual world entities management screen. To switch to this screen from the virtual world entities management screen the user must click in the navigation buttons (Topic/Key) located in the lower part of the user interface, as Figure 8 shows:



Figure 8. Navigation button of TaggigCreaditor.

When managing the virtual-world entities, we have three options for each: create entity, edit entity or delete entity. To manage virtual world entities, we will have to do on the screen I called virtual world management screen shown above. We will be seeing each of the cases of creation, and then I will explain the case of editing and deleting with an example, because it is applicable to all entities.

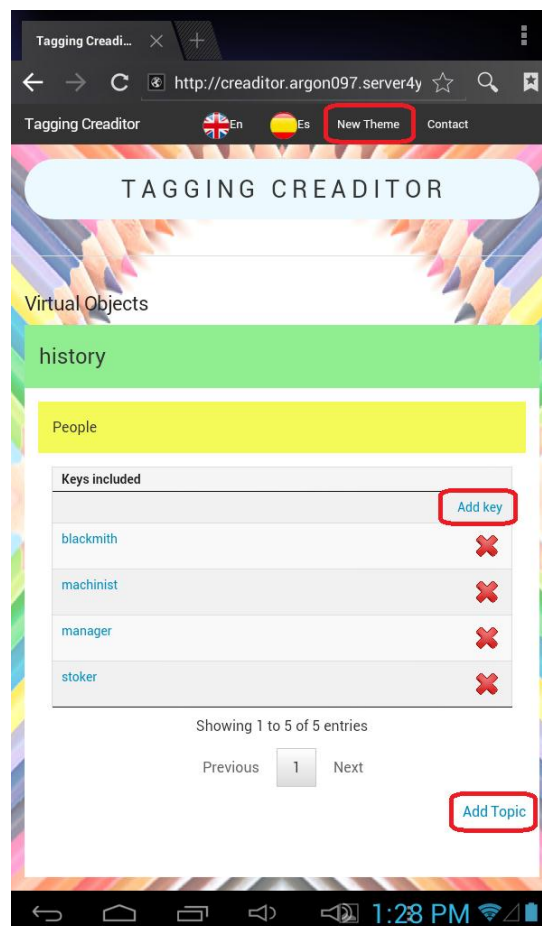


Figure 9. Virtual world management screen and its creation buttons.



### Creating a new theme

To add a new theme the user has to go to the top bar menu of the virtual world entities management screen. After selecting this option the user will see the creation screen of a theme. In it, as can be seen in Figure 10, the name field to create a new theme has to be filled or, as a second option, it is offered to the user to add an existing theme. As was the case of playspaces, if the user chooses to add an existing theme, the second screen of Figure 10 will be seen, where the user can choose either to add an existing or clone an existing one. The difference is that adding an existing one, by modifying features or adding objects to this theme, this theme in the other games will be modified (i.e., the same theme may be modified in whatever the game containing this theme); while cloning an existing one means that you copy all the features and objects found in the theme and are copied to the game, that is, if the user changes something in the cloned theme, it will not be changed in the origin theme.

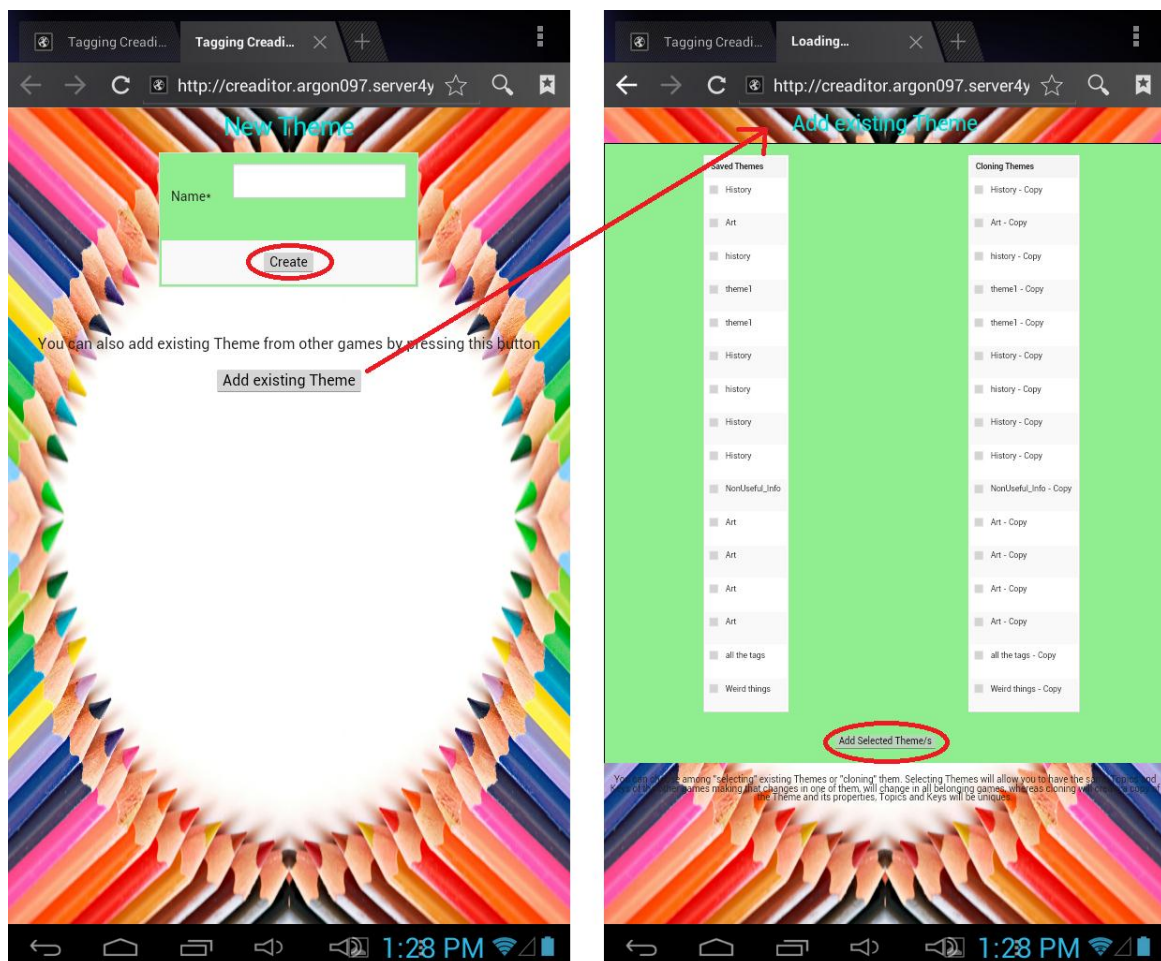


Figure 10. Creation a new theme or adding an existing one.

### *Creating a new topic*

To add a new topic to an existing theme, the user firstly needs to choose from the list of objects the theme to add the topic to create. By choosing it, the topics that are part of the theme, if any, and the option of adding a new one will appear. Once the user chooses the option to add topic, the creation screen of new topic appears. In it, the user must fill in the name and information fields and also choose a picture to identify the topic. After that, the topic will have been created.



Figure 11. Creation of a new topic.

### Creating a new key

To add a new key to an existing topic, the user firstly needs to choose from the list of objects the topic to add the key to create. By choosing it, a list will appear with the keys that are part of the topic, if any, and the option to add new one. Once the user chooses the option to add key, a box in which the data should be entered will be displayed. The field of information and, if user has previously added all the optional attributes, visibility and hint will also appear. After clicking the create button, the new key will be created.

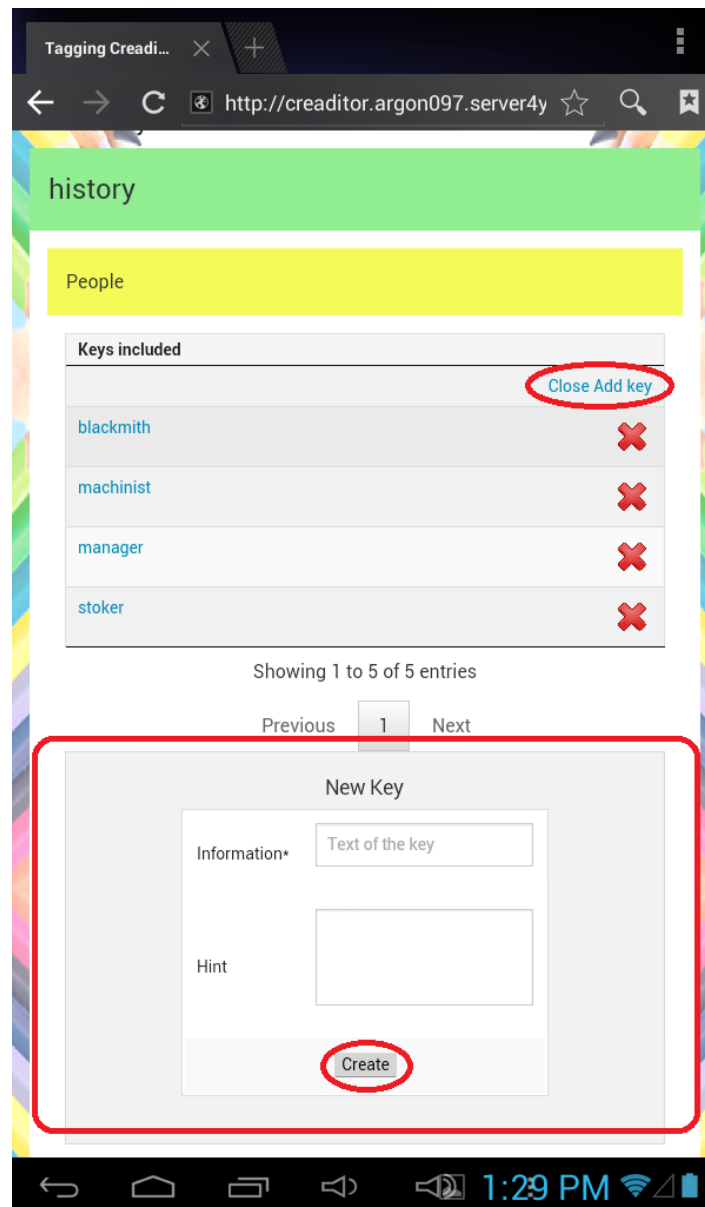


Figure 12. Creation of a new key.

*Editing and deleting virtual world entities*

The edition of virtual world entities is exactly like the process described for real world entities in the previous section.

To delete any of the entities in the virtual world the same process for real-world entities is followed and, as was the case for the elements, the application provides a faster way when removing keys: given the user the option to delete keys from the object navigation list.

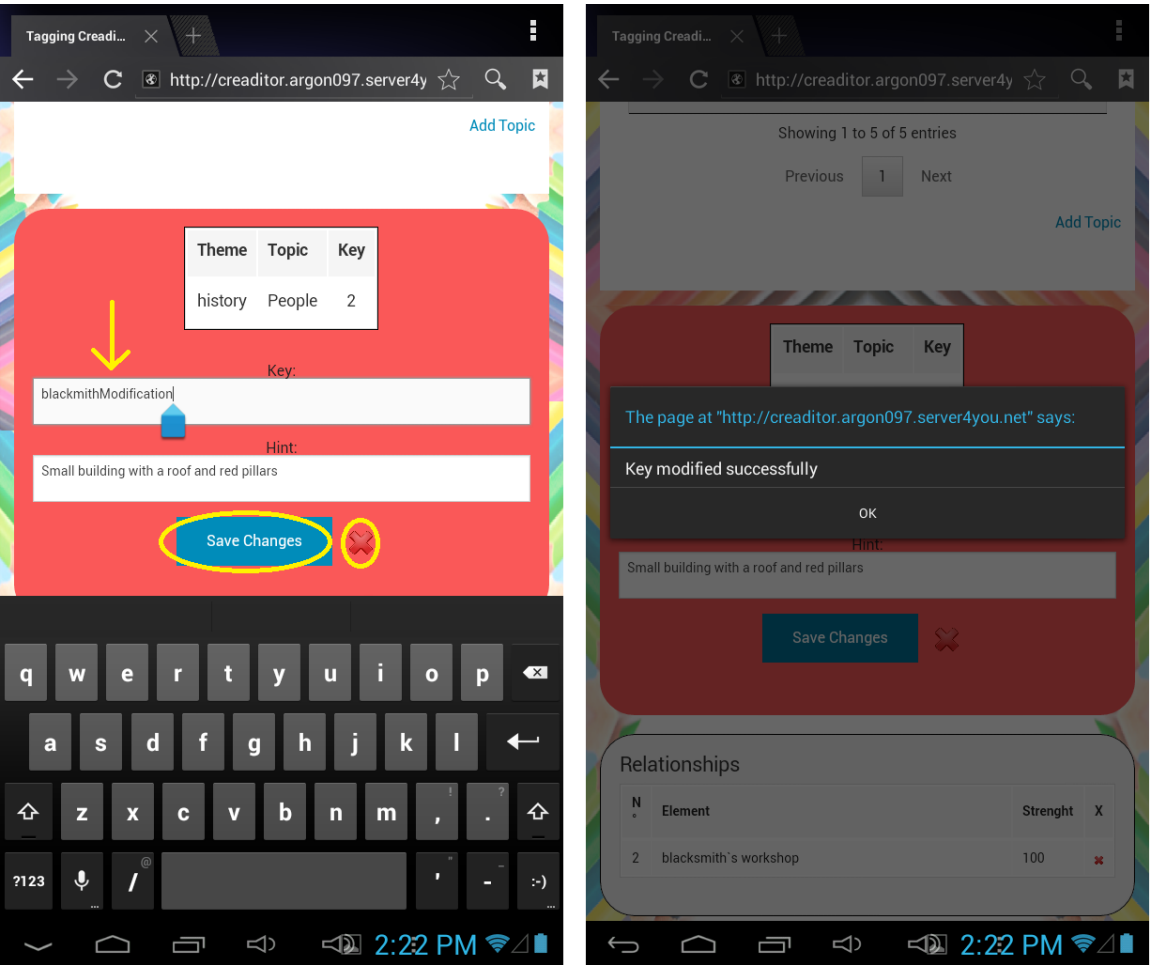


Figure 13. Editing a key as an example.



## V. Linking

Once you have created all necessary entities in both the real world and the virtual world for a game, you can use the linker to relate real world entities of third level with virtual world entities of third level, i.e., the elements and keys. The linker also provides an option to view all the created relationships. The user can go to the linking screen through the navigation buttons located in the lower part of the screen:



Figure 14. Editing a key as an example.

The process when making a relationship consists of three simple steps:

1. Choose the item you want to include in the relationship. This element is selected in the navigation list of physical entities.
2. Choose the key you want in the relationship. This key is selected in the navigation list of virtual entities.
3. Create relationship entering required attributes. In the example below, only the Strength is required.

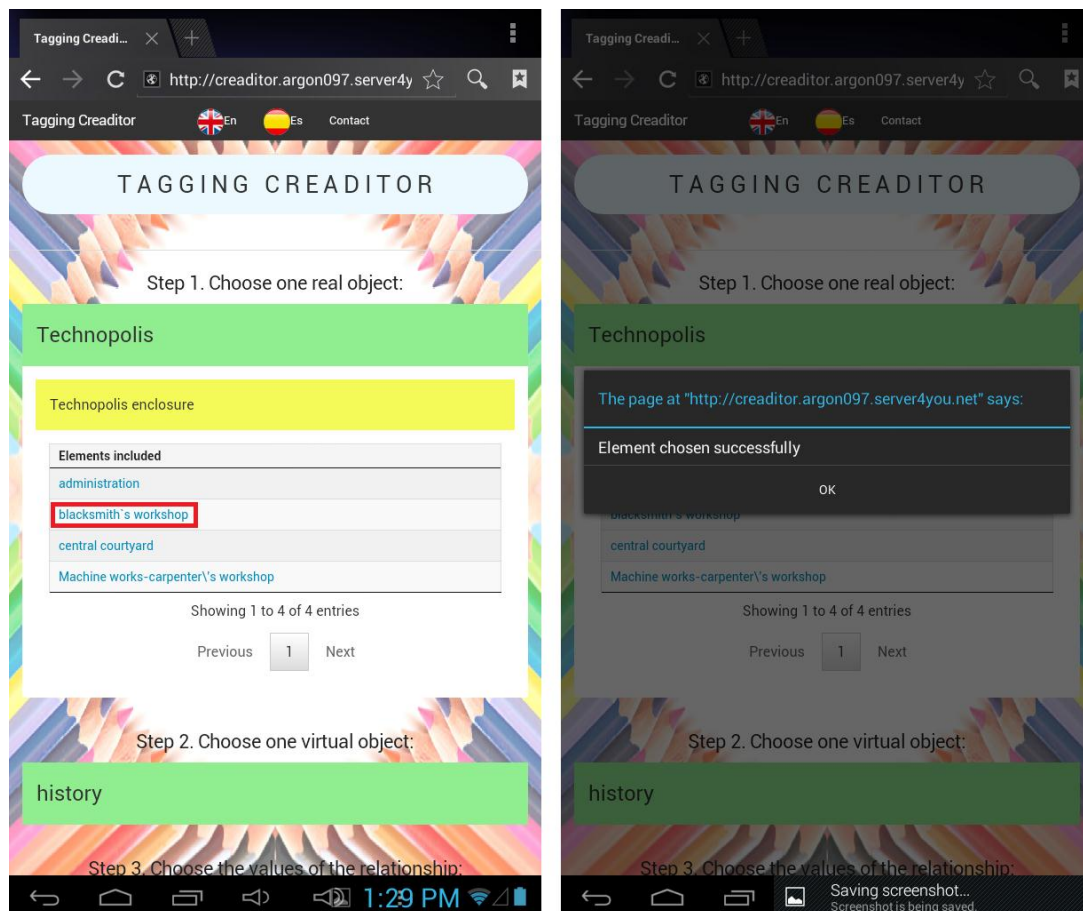


Figure 15. Step 1 of the creation of a relationship: select the element.

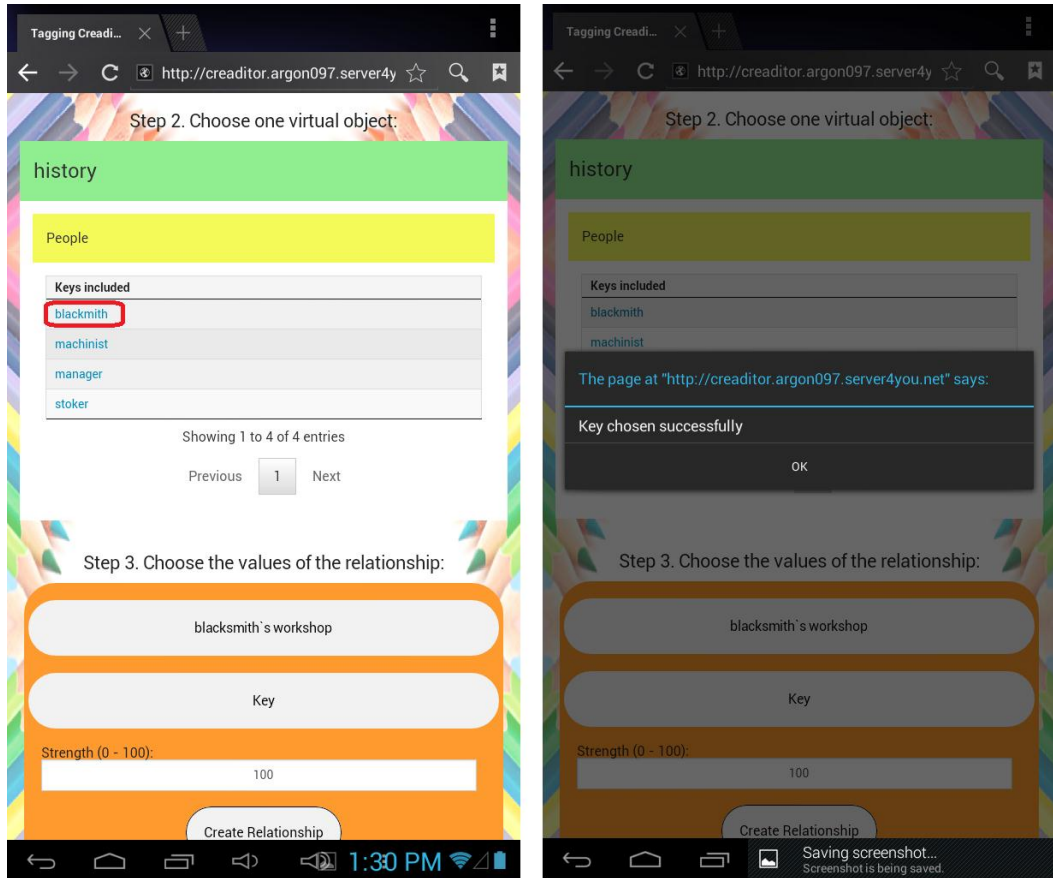


Figure 16. Step 2 of the creation of a relationship: select the key.

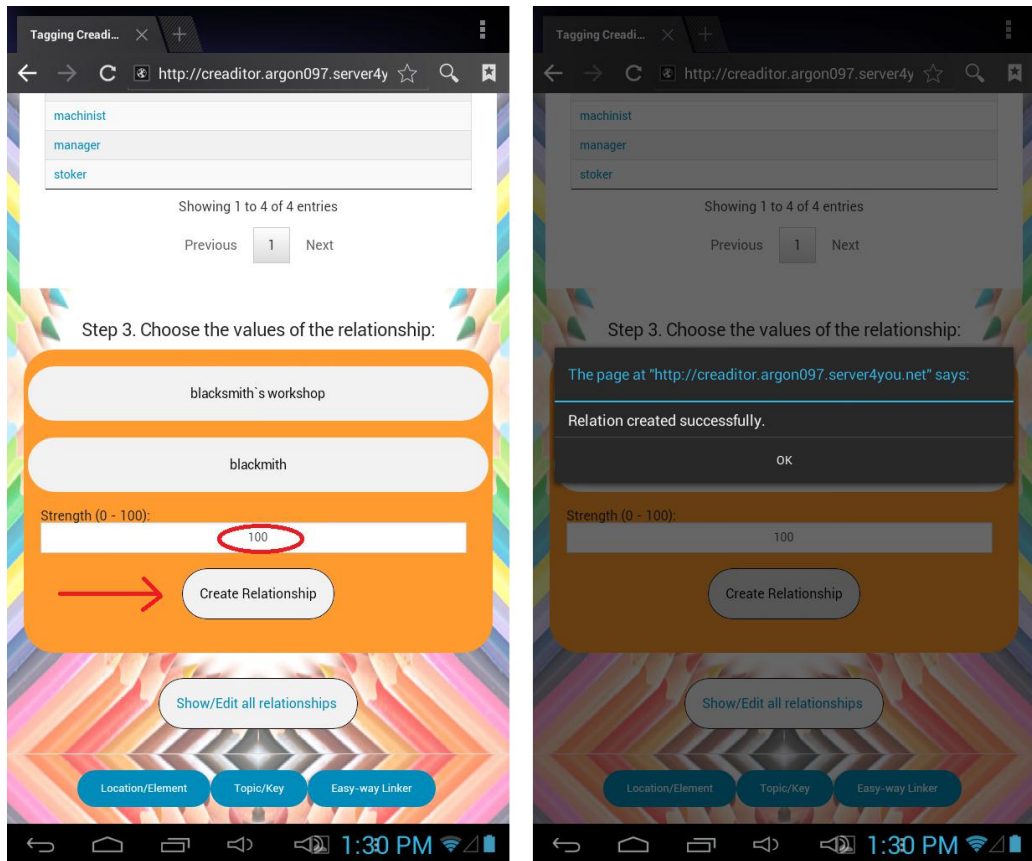


Figure 17. Step 1 of the creation of a relationship: enter attributes and finish.

When the user has finished the creation of relationships, it is possible to check if all the relationships were created properly. The user can do this by clicking the button 'Show/Edit all relationship' located in the lower part of the linking screen (Figure 17).

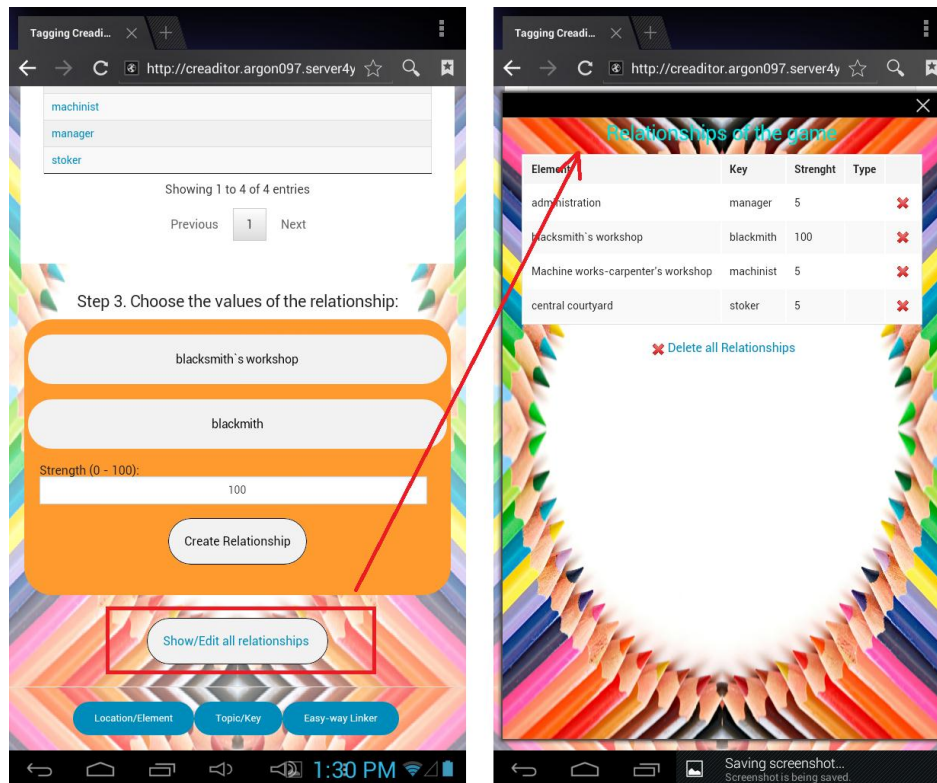


Figure 18. The user can check all the created relationships.

## VI. Create content for an existing game

Once the common scenario has been explained, another common scenario will be explained: The user wants to create or modify content for an existing game. To do that, the user must choose the option 'Select game' in the Welcome screen and the game list screen will appear. In it, the user can choose a game in the list and see the details of it in the detail box. When the desired game is chosen, the following step is to create or modify the content that has already been created. To do it, the user must click in the 'Create/Edit content for this game' button as it is showed in the Figure 19. After clicking it, the user will see the real world management screen and the steps are exactly the same as the ones explained in the new game scenario.

Before finishing the manual, it has not been said that the application supports two languages: English and Spanish. Users can change it whenever they want in the top bar menu of all the screens of the game.

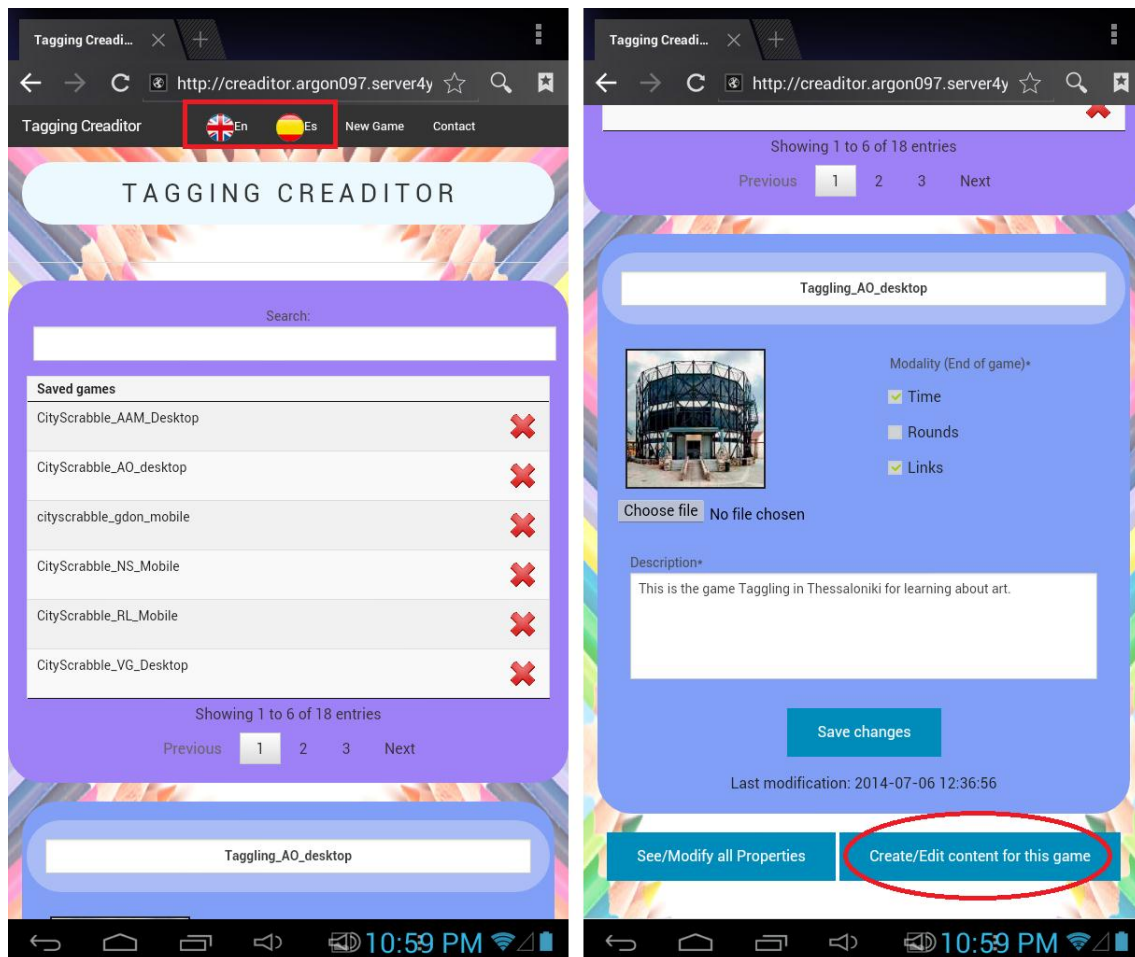


Figure 18. Creating content for an existing game and changing the language.

## ANNEX 2. EVALUATION SESSION DOCUMENTS<sup>20</sup>

### 1. Evaluation of TaggingCreaditor

#### PERSONAL INFORMATION

Name (Not Compulsory): .....

Surnames (Not Compulsory): .....

Age:..... Sex: ☐ Male ☐ Female

Nationality:.....

Education: .....

#### INFORMATION FOR TESTERS

All data and information got from this evaluation will be used for the evaluation and analysis of the developed tool, TaggingCreaditor. This information will be anonymous and the results will be only used as part of Rodrigo's and Alejandro's thesis. By completing this form, tester agrees that he/she can be recorded during the whole process. It comprises the understanding, use and explanation of how to use TaggingCreaditor.

#### INITIAL INFORMATION

TaggingCreaditor is a content provider for location-based games which allows for non-professional people to create their own content for already developed games and even for the new ones, in an easy way. Some of these already developed games are CityScrabble and Taggling, which you will understand and create content for them later. TaggingCreaditor has been developed for desktop and mobile devices. So tester will have to try the working for both of them in the two different scenarios.

During the evaluation, the tester will have to follow the 'Think-Aloud Protocol', which consists of observing a user working with an interface while encouraging them to "think-aloud"; to say what they are thinking and wondering at each moment. This protocol is of particular value because it focuses on the problems a user has; when the user is working without difficulty, direct observation (and hence the think-aloud protocol itself) is of very limited use. Therefore, following this protocol, you will have to verbalize and talk all the actions that you are going to do while you are using TaggingCreaditor.

The structure of this evaluation is the following:

1. First of all, understanding the performance of the games which new content is needed.
2. We will describe two scenarios for each game, one for desktop and one for mobile version.
3. Then you will create the new content following some guidelines.
4. Finally, you will answer a short questionnaire about the application.

---

<sup>20</sup> Documents performed with the help of Alejandro Ortega Arranz.



# CITYSCRABBLE

## HOW TO PLAY

CityScrabble is a location-based game for open areas. It is played using android devices in places for which content has been prepared.

The game can be played in any open space. Examples are city centers, archaeological parks or other sites of some importance. It can also be played in closed spaces such as museums or any other relative large space.

In CityScrabble each player (or team) tries to locate locations or objects in space and connect them with keys (small sentences). In order to do that, she has to go near them (to trigger a gps proximity sensor) or scan a QR tag. When a location is identified, it can be connected with any key. Points are only awarded if the connection between key and the object is meaningful.

It is a competitive multiplayer game. It can be played by competing individuals or by teams of 2-3 players. It should last a little more than 30 minutes, depending on the size of the area.

The game consists of three screens, the topic selection screen, the association screen and the map screen.

### 1. Topic selection screen

When the game starts, the player sees the topic selection screen with 9 topics and each contains 4 keys.

Each topic icon has four flags signaling availability of its keys. The keys take the color of the team which has used them in a connection. At the instance below, Mae's color is purple and the player with the red color has already connected a key.

The player now has to look around in the game space (playspace) and try to locate real objects (elements) such as exhibits, sculptures, paintings... that might be relevant to the chosen topic and the available keys. The way to do this is to search in the game space for the real objects and scan their tag with the handheld device.



## 2. Association screen

Having chosen a topic, the exhibit association screen is displayed. The player sees a short text describing the concept of the topic and below, a list of up to four keys. Initially, they are not connected. During the game they take the color of the player who used them in a combination. Once a real object is tagged, the scanned exhibit is then included in one of the available slots.

At the bottom of the screen are the objects that have been found or the empty sockets for the objects that are still to be found. By pressing the lock at the middle, an object is connected or unconnected to a key. Each key explains something related to one of the real objects, so that connections have to be meaningful.



Once the slot is filled with an object, the link/unlink icon appears and the player can link this object with the selected key by pressing the link/unlink icon. When a player links a key, this key is flagged with the player's color and is marked as taken. This means that the key is no longer available for any other player. Moreover, if the relationship between the key and the object is right, the player gets points.

The player can continue looking in the game space for objects that match the current topic or at any time return to the topic selection screen. When the player returns to the topic screen, it is visible that the other teams have been also making relationships. This gives an overview of the other player's activity, but doesn't provide any information about their score.

## 3. Map Screen

A map assists the players by showing the position of the objects on the field. The not yet found objects are marked with a question mark.



## SCENARIO 1 – DESKTOP

You are a history teacher in a school of Patras and your students and you are going to the Technopolis Museum in Athens and Museum of Contemporary Art in Thessaloniki next week. The best way for your students to learn knowledge about it is playing CityScrabble (Technopolis) and Taggling (Contemporary Art) with your own content. You are right now at home because you already have all the information about them stored in your computer. What you have to do, then, is to add content with the TaggingCreaditor editor uploading the information you have to the application. The steps to do so are as follows:

- 1-Create a new game.
- 2-Enter real data. Put the data of the playspaces, locations and elements in the Location/Element screen (the one appearing after creating a game).
- 3-Enter virtual data: Put the data of the themes, topics and keys in the Topic/Key screen.
- 4-Create the relationships between the elements and the keys in the Easy-way Linker screen. Check that the relationships have been created properly.

The data you have to introduce can be found in the annexes 'CityScrabble-Instructions for the desktop scenario' and 'Taggling-Instructions for the desktop scenario', respectively.

## SCENARIO 2 – MOBILE

You are a history teacher in a school of Patras and your students and you are going to the Technopolis Museum in Athens Athens and Museum of Contemporary Art in Thessaloniki next week. The best way for your students to learn knowledge about it is playing CityScrabble with your own content. Imagine that you are right now in the Technopolis Museum in Athens -or in the Contemporary Art Museum in Thessaloniki- with the purpose of creating content for these games with a tablet. What you have to do, then, is to add content with the TaggingCreaditor editor at the same time as you go around the museum. The steps to do so are as follows:

- 1-Create a new game.
- 2-Enter real data. Put the data of the playspaces, locations and elements in the Location/Element screen (the one appearing after creating a game).
- 3-Enter virtual data: Put the data of the themes, topics and keys in the Topic/Key screen.
- 4-Create the relationships between the elements and the keys in the Easy-way Linker screen. Check that the relationships have been created properly.

The data you have to introduce can be found in the annexes 'CityScrabble-Instructions for the mobile scenario' and 'Taggling-Instructions for the mobile scenario', respectively.



## QUESTIONS ABOUT THE DESKTOP APPLICATION

1 = Strongly disagree

5 = Strongly agree

Regarding this questionnaire, please, remember that you play a role so that you should answer as if you are the teacher of the scenario described above.

QUESTION	1	2	3	4	5
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technical person to be able to use this system.					
5. I found the various functions in this system were well integrated.					
6. I thought there was too much inconsistency in this system.					
7. I would imagine that most people would learn to use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. I needed to learn a lot of things before I could get going with this system.					

## QUESTIONS ABOUT THE MOBILE APPLICATION

1 = Strongly disagree

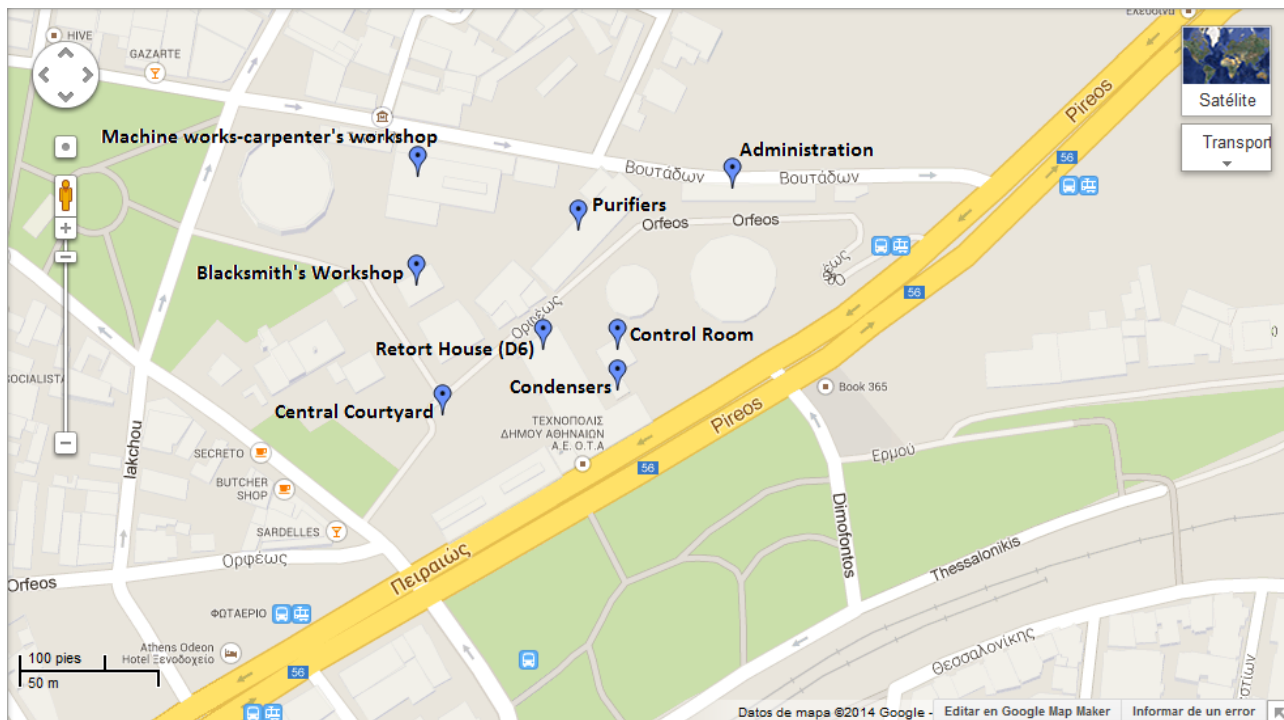
5 = Strongly agree

Regarding this questionnaire, please, remember that you play a role so that you should answer as if you are the teacher of the scenario described above.

QUESTION	1	2	3	4	5
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technical person to be able to use this system.					
5. I found the various functions in this system were well integrated.					
6. I thought there was too much inconsistency in this system.					
7. I would imagine that most people would learn to use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. I needed to learn a lot of things before I could get going with this system.					

## 2. CityScrabble - Instructions for the mobile scenario

Below is a list of buildings or other kinds of entities that should be included in a game of CityScrabble. The list contains descriptions of buildings or other entities. Here we have the location of the Elements to introduce. For it, once you are entering the Element information, you have to look for "Technopolis Athens" in the map, and then drag the cursor until the right place:



### New Game

<b>Name</b>	<b>CityScrabble_NS_Mobile</b> (N: 1st name letter, S: 1st surname letter)
<b>Version</b>	<b>1</b>
<b>Description</b>	<b>Technopolis game for the students of Patra's school.</b>
<b>Language</b>	<b>Greek</b>
<b>Modality</b>	<b>Rounds</b>
<b>Game image</b>	<b>No image</b>
<b>Hierarchy</b>	
<b>Objects</b>	<b>Use the default names</b>
<b>Name</b>	
<b>Plugging</b>	<b>For CityScrabble, we need:</b>
<b>Properties</b>	<b>Element --&gt; Radius : default value: 10m</b>
<b>Block</b>	<b>Key --&gt; Hint: without default value</b>

## Real Objects (Location/Element screen)

<b>Playspace</b>	<b>Technopolis</b>
<b>Kind</b>	<b>Museum</b>
<b>Information</b>	<b>Industrial museum in of Athens. It is situated in the city's former gasworks (factory for the production of industrial gas) which were founded in 1857.</b>

<b>Location</b>	<b>Technopolis Enclosure</b>
<b>Type</b>	<b>Outdoor</b>
<b>Information</b>	<b>Main yard of the museum.</b>

Now, go around the museum to look for the eight elements located in the ‘Technopolis Enclosure’ location. When you find an element, you will have to enter it in the application, taking into account that you will have to take a photo of the element with the tablet (horizontally).

## Virtual Objects (Topic/Key screen)

NOTE: In order to see the keys that have been added to a topic, you have to refresh the page. You can create all the keys in a row without refreshing the page and, when you finish, you can refresh the page to see the result.

<b>Theme</b>	<b>History</b>
--------------	----------------

<b>Topic</b>	<b>1</b>
<b>title</b>	<b>People</b>
<b>image</b>	<b>topic1.jpg (In the gallery of the tablet)</b>
<b>description</b>	<b>People of different specialties used to work in the Athens gasworks. Try to locate in which building some of them worked.</b>

Info key	Manager
Hint key	The boss who supervised everything had his own building, near one of the entrances.

Info key	Blacksmith
Hint key	Small building with a roof and red pillars.

Info key	Machinist
Hint key	The largest building in Technopolis

Info key	Stoker
Hint key	Venue for concerts

<b>Topic</b>	<b>2</b>
<b>title</b>	<b>The production line in the gasworks</b>
<b>image</b>	<b>topic2.jpg (In the gallery of the tablet)</b>
<b>description</b>	<b>The gas was produced in many stages. Find the most important ones.</b>

Info key	Overheating
Hint key	A gas is born in an oven

Info key	Cooling
Hint key	The newborn needs cooling down in a lot of pipes

Info key	Purification
Hint key	Sulfide

Info key	Distribution
Hint key	Master control in a small building

## Relationships (Easy-way linker screen)

<b>Topic</b>	<b>1</b>
--------------	----------

Info Key	Manager
Element	Administration
Score	5

Info Key	Blacksmith
Element	Blacksmith's Workshop
Score	5

Info Key	Machinist
----------	-----------

Element	Machine works-Carpenter's workshop
Score	5

Info Key	Stoker
Element	Central Courtyard
Score	5

<b>Topic</b>	<b>2</b>
--------------	----------

Info Key	Overheating
Element	Retort House (D6)
Score	5

Info Key	Cooling
Element	Condensers
Score	5

Info Key	Purification
Element	Purifier
Score	5

Info Key	Distribution
Element	Control Room
Score	5

### 3. Taggling - Instructions for the mobile scenario

#### New Game

<b>Name</b>	<b>Taggling_NS_mobile</b> (N: 1st name letter, S: 1st surname letter)
<b>Version</b>	<b>1</b>
<b>Description</b>	<b>Taggling game for the students of Patra's school.</b>
<b>Language</b>	<b>Greek</b>
<b>Modality</b>	<b>Relationships</b>
<b>Game image</b>	<b>No image</b>
<b>Hierarchy</b>	
<b>Objects</b>	<b>Use the default names except "Tag" instead "Key"</b>
<b>Name</b>	
<b>Plugging</b>	
<b>Properties</b>	<b>We do not need extra properties</b>
<b>Block</b>	

#### Real Objects (Location/Element)

<b>Playspace</b>	<b>Contemporary Art Museum</b>
<b>Kind</b>	<b>Museum</b>
<b>Information</b>	<b>Contemporary Art Museum in Thessaloniki.</b>

<b>Location</b>	<b>Museum Rooms</b>
<b>Type</b>	<b>Indoor</b>
<b>Information</b>	<b>The elements are located in the different rooms of the museum.</b>

Now, go around the museum to look for the eight elements located in the 'Museums rooms' location. When you find an element, you will have to enter it in the application, taking into account that you will have to take a photo of the element with the tablet (horizontally).

**IMPORTANT:** Take into account that this game does not use location in maps. That is, when you are introducing an element in the application, you should leave the default value of the location.

## Virtual Objects (Topic/Tag)

<b>Theme</b>	<b>Art</b>
--------------	------------

<b>Topic</b>	<b>1</b>
<b>title</b>	<b>Contemporary Art</b>
<b>image</b>	<b>No image</b>
<b>description</b>	<b>Contemporary art is art produced at the present period in time.</b>

----- Relationships with Element 1 -----

Info key	Music Sculpture
----------	-----------------

Info key	Kinetic Art
----------	-------------

Info key	Art and Science
----------	-----------------

Info key	Chord
----------	-------

Info key	Sound Box
----------	-----------

Info key	Invisible Energy
----------	------------------

----- Relationships with Element 2 -----

Info key	Sculpture crease
----------	------------------

Info key	Theatricality
----------	---------------

Info key	Presence
----------	----------

Info key	Absence
----------	---------

Info key	Loneliness
----------	------------

Info key	Toughness
----------	-----------

----- Relationships with Element 3 -----

Info key	War
----------	-----



Info key	Casket
----------	--------

Info key	Innocent victims
----------	------------------

Info key	Childhood
----------	-----------

Info key	Death
----------	-------

----- Relationships with Element 4 -----

Info key	Body
----------	------

Info key	Couple
----------	--------

Info key	Adam and Eve
----------	--------------

Info key	Distortion
----------	------------

Info key	Sex
----------	-----

Info key	Phallus
----------	---------

----- Relationships with Element 5 -----

Info key	Popular art
----------	-------------

Info key	Image
----------	-------

Info key	Historical currency
----------	---------------------

Info key	Historical figure
----------	-------------------

Info key	Face
----------	------

Info key	Symbol
----------	--------

----- Relationships with Element 6 -----

Info key	Moving sculpture
----------	------------------

Info key	Macabre puppet
Info key	Skeleton
Info key	Dance
Info key	Repetition
----- Relationships with Element 7 -----	
Info key	Out of frame
Info key	Absence
Info key	Fluo colors
Info key	Chicken

### Relationships (Easy-way Linker)

- The relationships have been marked in the previous section.
- In that game, all the relationships have the same default score: 100.
- Moreover, some of the tags are related with two different elements, that is why you should add these relations more:
  1. Death with Element 6
  2. Repetition with Element 7
  3. Face with Element 7